데이터 복구의 거의 모든 것

JK Kim

@pr0neer

forensic-proof.com

proneer@gmail.com

개요

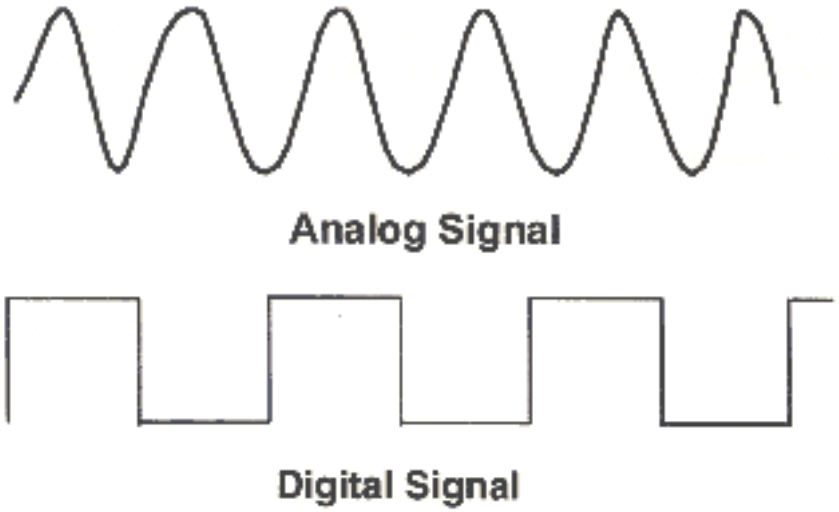
1. 데이터 기록 방식
2. 데이터 인코딩 기법
3. 물리적 손상 복구
4. 논리적 손상 복구
5. 삭제된 파일 복구
6. 데이터 영구 삭제 기법
7. 영구 삭제 후 복구 가능성

데이터 기록 방식

- 자기 기록 (Magnetic Recording)
- 광학 기록 (Optical Recording)
- 전자 기록 (Electronic Recording)

기본 개념

- **데이터(Data)** (pron.: /ˈdeɪtə/ day-tə or /ˈdætə/)
  - the **quantities**, **characters**, or **symbols** on which operations are performed by a computer, being stored and transmitted **in the form of electrical signals** and recorded on **magnetic**, **optical**, or **mechanical recording media** – *Wikipedia*
  - In computer science, data is anything **in a form suitable for use with a computer** – *Wikipedia*



저장매체 기록 방식

- **자기 기록 (Magnetic Recording)**
  - 하드디스크, 마그네틱 테이프, 마그네틱 카드(신용카드) 등
- **광학 기록 (Optical Recording)**
  - CD(Compact Disk), DVD(Digital Versatile Disk), 블루레이(Blu-ray) 등
- **전자 기록 (Electronic Recording)**
  - 플래시 메모리(Flash Memory), SSD, 램(RAM), 롬(ROM) 등

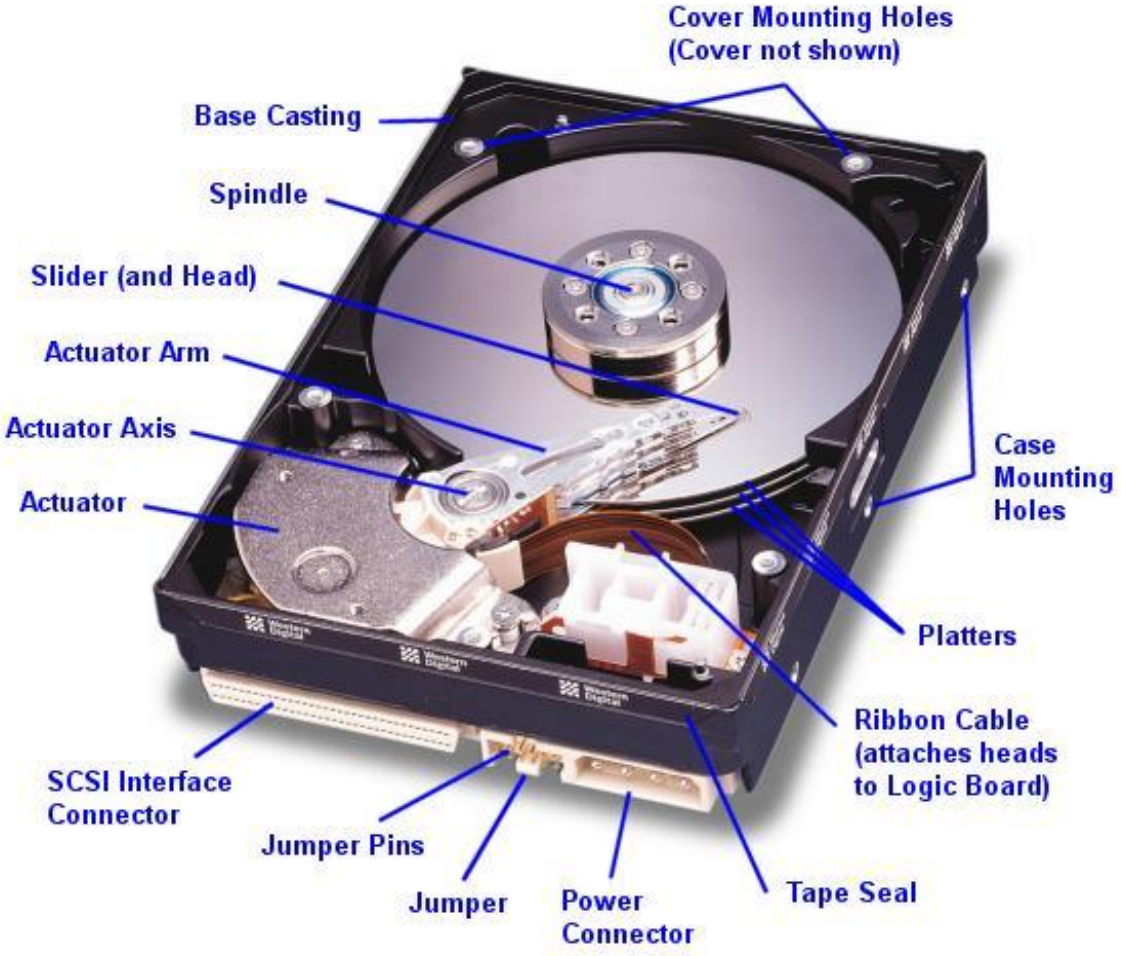
자기 기록

- 하드디스크 드라이브(HDD) 내부

http://www.hitechreview.com/pictures/western-digital-caviar-green-hdd/

자기 기록

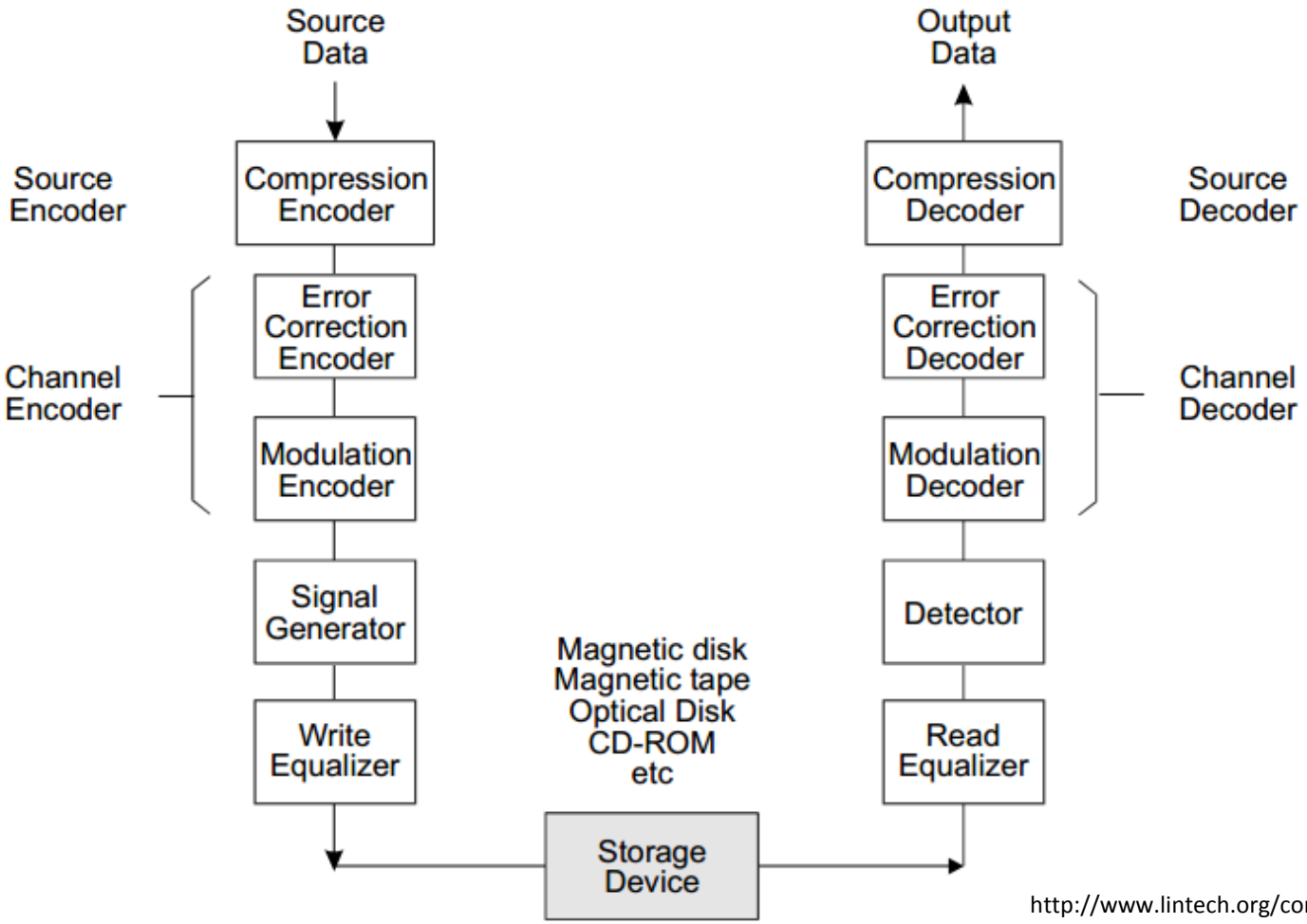
- HDD 주요 구성 요소



http://datarecoverydoctor.co.uk/images/hdd-diagragm-parts-guide.gif

자기 기록

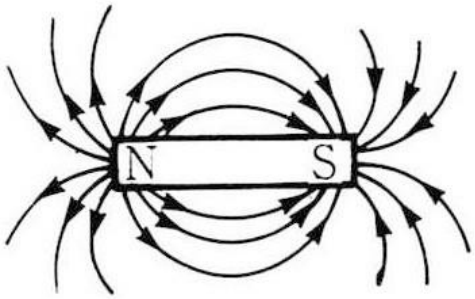
- 디지털 데이터 읽기/쓰기 채널



http://www.lintech.org/comp-per/07MAGREC.pdf

자기 기록

- **자성체 (magnetic substance) ➔ 자기장 내에서 자화되는 물질 (지구상의 모든 물질)**



- **자성체 분류**
  - **강자성체(ferromagnetic substance)** : 외부의 강한 자기장이 있을 때, 그 자기장 방향으로 강하게 자화된 뒤 자기장이 사라져도 자화가 남아있는 물질 (철, 코발트, 니켈 등)
  - **반자성체(diamagnetic substance)** : 외부 자기장에 의해 자기장과 반대 방향으로 자화되는 물질 (금속과 산소를 제외한 기체, 물 등)
  - **상자성체(paramagnetic substance)** : 자기장안에서는 자기장 방향으로 약화게 자화하고, 자기장이 제거되면 자화하지 않는 물질 (알루미늄, 주석, 백금, 이리듐 등)

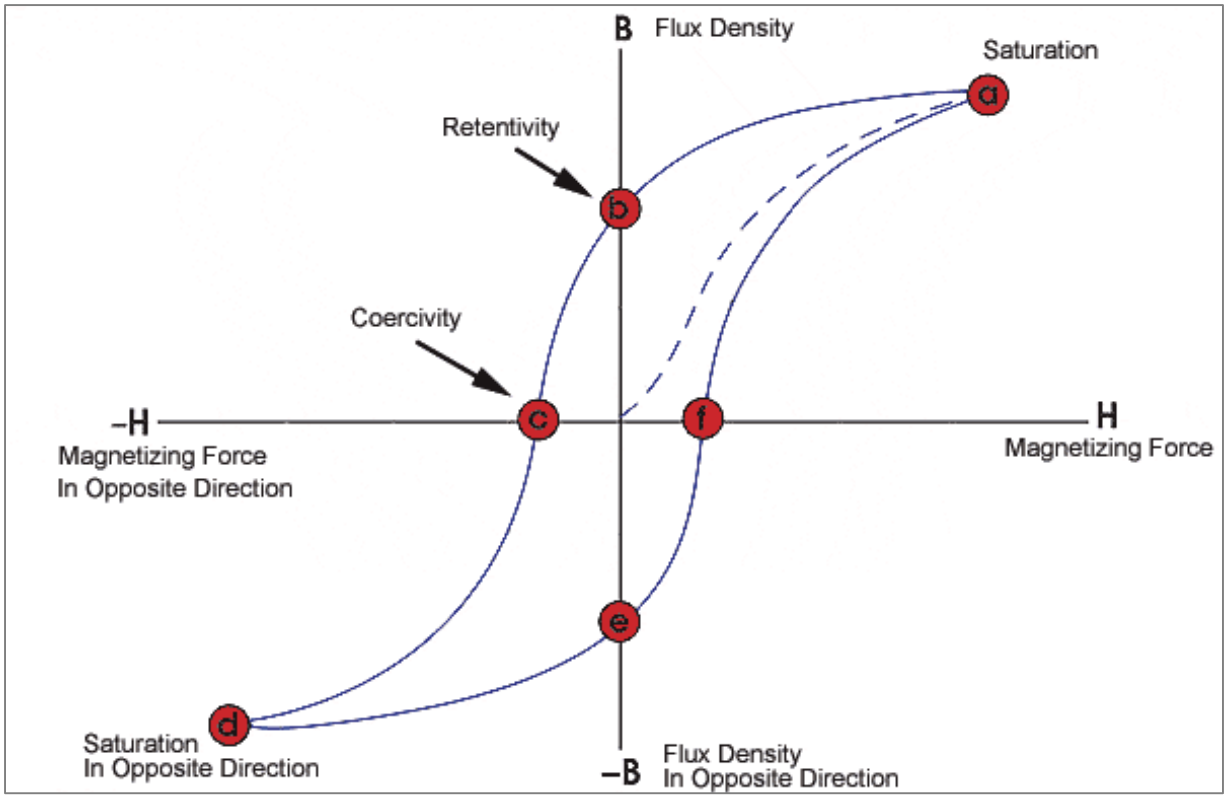
자기 기록

- **자성체 특징**

  - **포화 상태(Saturation)** : 자화되지 않은 상태에서 자기장 세기(H)를 증가시켰을 때 더 이상 증가하지 않는 상태
  - **잔류 자기(Retentivity, Remanence)** : 자기 포화상태에서 자화를 감소하여, 자기장를 제거했을 때 자성체에 남아있는 자력, 강자성체는 잔류 자기가 남기 쉬움 (영구 자석)
  - **보자력(Coercivity) :** 잔류 자기를 없애기 위해(0으로) 역방향으로 가해야 하는 자기장의 세기
  - **자속 밀도(magnetic flux density)** : 자기장의 크기를 나타내는 것으로 단위 면적을 수직으로 지나는 자기력선의 수
  - **자기력(magnetic force)** : 두 극 사이에 작용하는 힘으로 서로 밀거나 당기는 힘

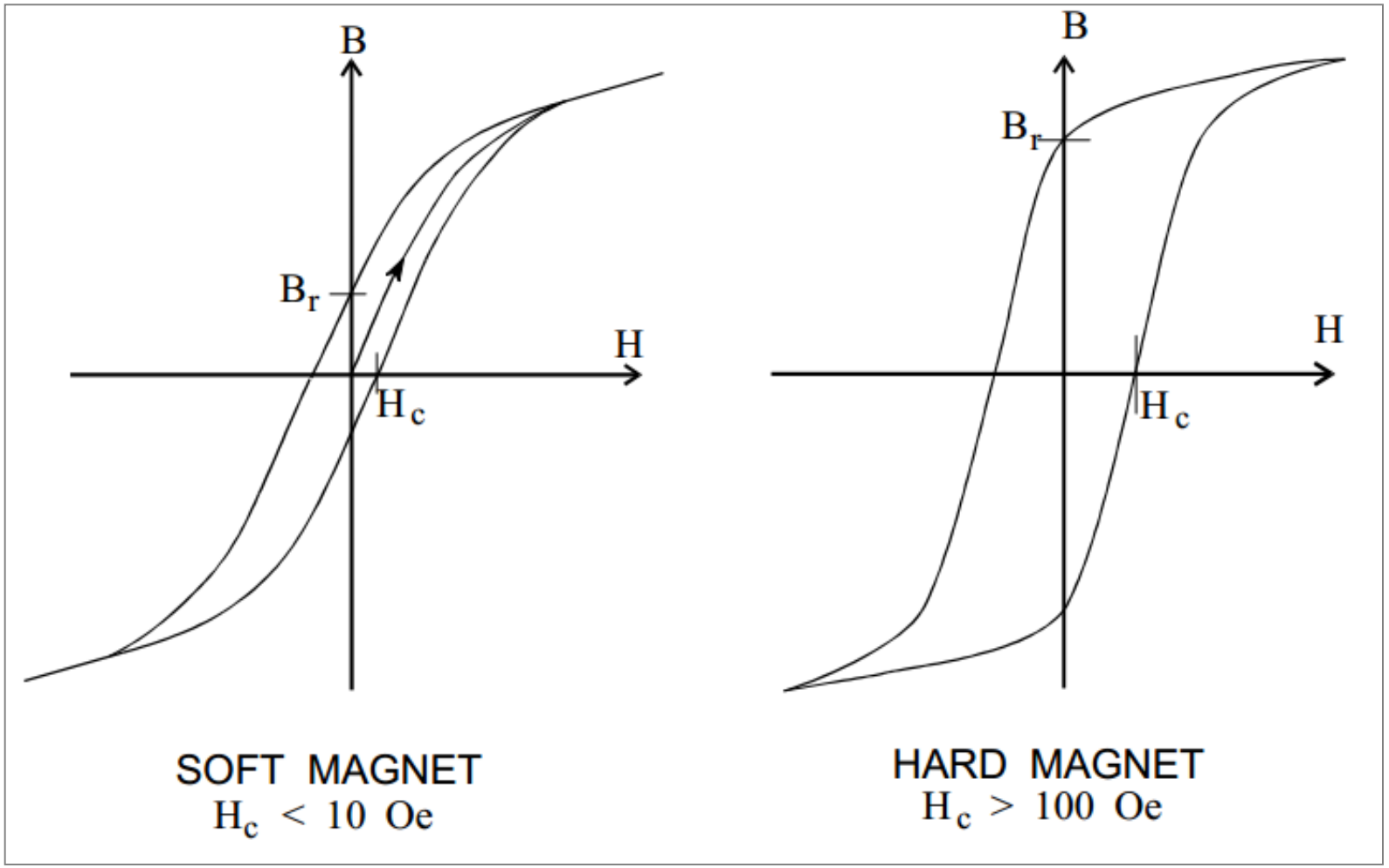
자기 기록

- **자기 이력 곡선 (Hysteresis Loop), B-H 곡선**
  - 디스크의 경우, 전압 제한, 타이밍, 쓰기 밀도 등으로 인해 포화 상태에 이르지 못함
  - 온도, 덮어쓰기 같은 요인으로 인해 각 쓰기 작업 마다 B-H 곡선이 다름



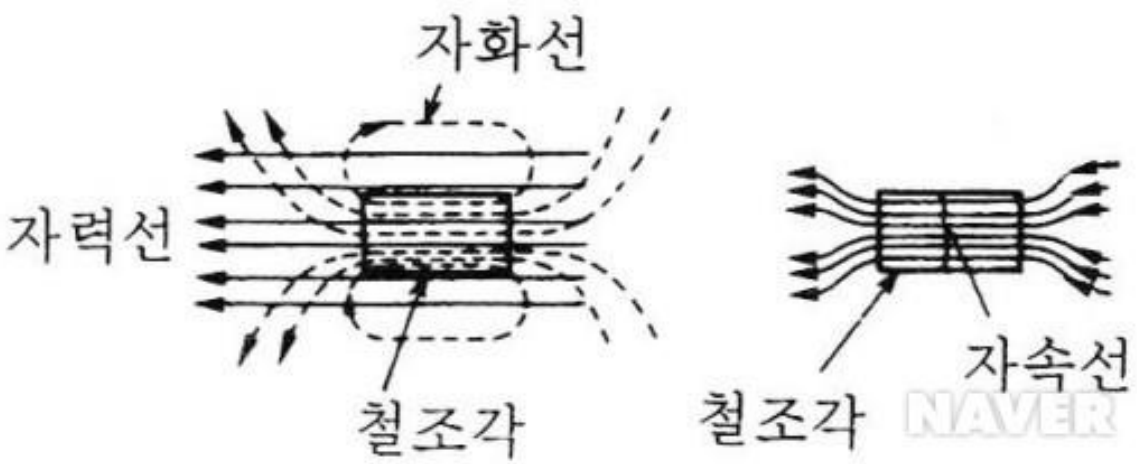
자기 기록

- 자기 이력 곡선 (Hysteresis Loop), B-H 곡선



자기 기록

- **자화 (magnetization) vs. 자속 (magnetic flux)**
  - **자화**
    - ✓ 자기장 내에 자성체를 놓았을 때, 해당 자성체도 자기를 띄게 되는 현상
  - **자속**
    - ✓ 자화된 자성체를 통과하는 자속선을 하나로 합한 총 자속선 수



데이터 기록 방식

자기 기록

- **0, 1 표현 방법 ➔ 두 극성을 활용 (North, South)**
  - **자극(magnetic pole) 측정을 이용한 방식**
    - ✓ N 극(또는 S 극) : 0
    - ✓ S 극(또는 N 극) : 1
  - **자화 반전(flux reversal)을 이용한 방식**
    - ✓ S → N(또는 N → S) : 0
    - ✓ N → S(또는 S → N) : 1
  - **절대적인 극성이 아닌 자화 반전을 측정하는 방식으로 발전 ➔ 왜???????**

자기 기록

- **자화 반전을 사용하는 이유**

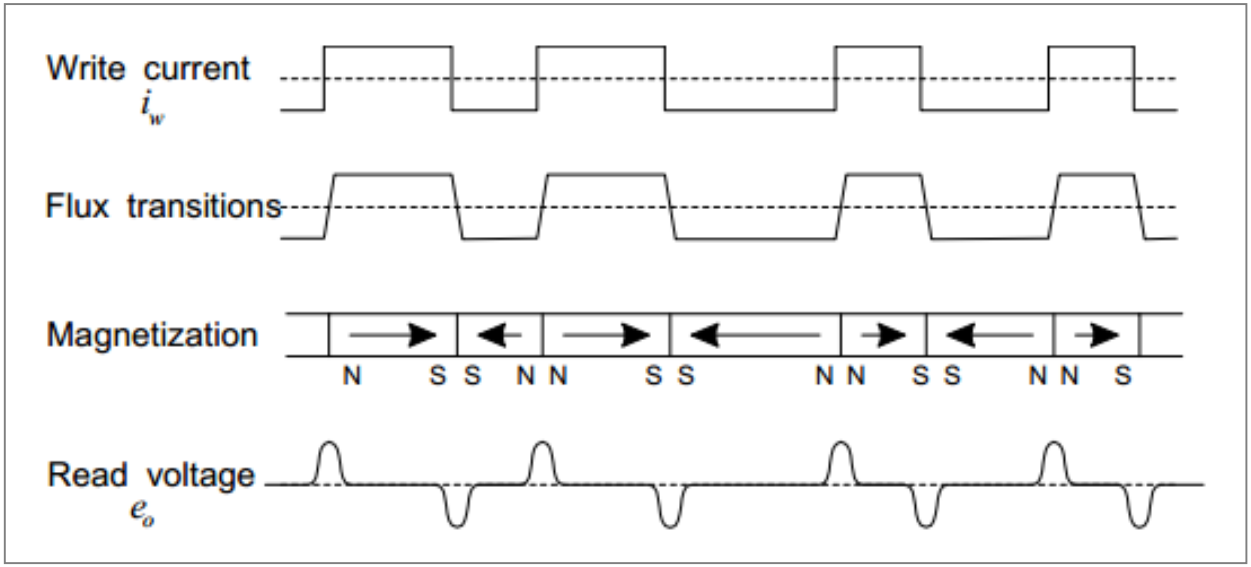
  - **필드(Field) vs. 반전(Reversal)**
    - ✓ 각 필드 값(극성)보다 반전을 측정하기가 더 쉬움
    - ✓ 반전을 자기저항 소자로 감지 ➔ 전압으로 변환 ➔ 헤드를 이용해 탐지
    - ✓ 디스크 밀도가 증가함에 따라 임계 감도를 탐지할 수 있는 자기장 세기는 계속 감소

  - **동기화(Synchronization)**
    - ✓ 각 비트의 시작과 끝이 어디인가?
    - ✓ 연속된 0을 1,000개 정도 기록할 경우 ➔ SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....

  - **필드 구분(Field Separation)**
    - ✓ 동일한 데이터(연속된 0혹은 1)가 기록될 경우, 하나의 큰 자기장 생성
    - ✓ 각 필드의 구분이 필요 ➔ 추가적인 필드 구분자가 필요함

자기 기록

- **기록에 따른 데이터, 자속 전이, 자화 흐름, 읽기 전압**



- 자성체에 외부 자계를 가하면 특정 방향으로 자화
- 자화 반전의 역할은 데이터를 구분하여 클록 동기화(clock synchronization)
- 클록 동기화에 사용되는 자화 반전을 어떻게 줄이냐? ➔ 인코딩 기법의 발달

데이터 기록 방식

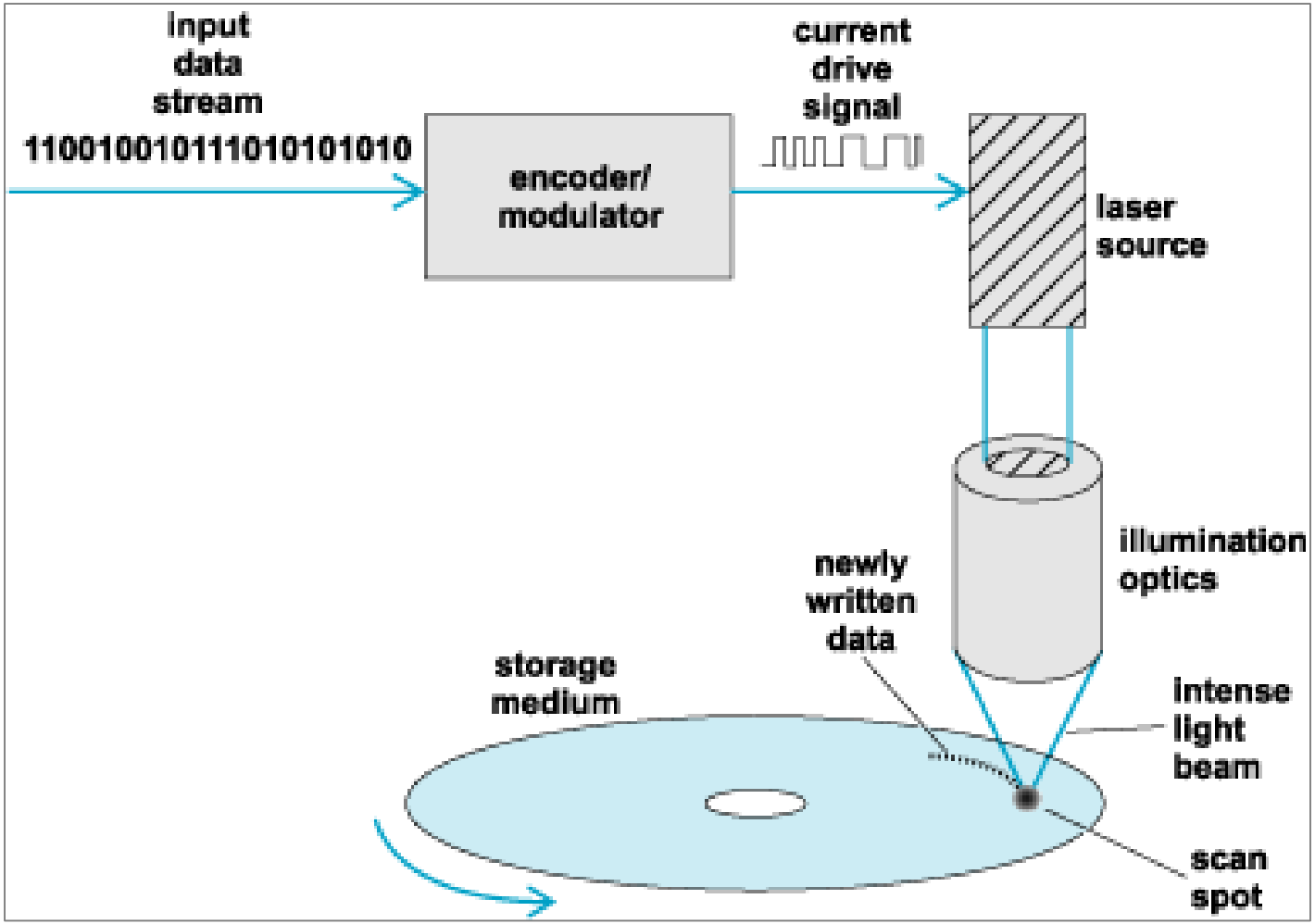
광학 기록

- **광학 필름 기록 (optical film recording)**
  - 동영상 기록(motion picture recording), 사진 기록(photographic recording)
  - 1970년대 후반까지 사진 필름 (photographic film)에 빛을 투사하여 신호(signal) 기록

- **레이저빔 필름 레코딩 (laser-beam film recording)**
  - 1970년대 후반 이후 레이저를 이용한 기록 방식 발달 ➔ 신호 기록율 향상
  - 펄스 부호 변조 (Pulse-code Modulation) 기법을 통해 고성능의 음성 신호 재생

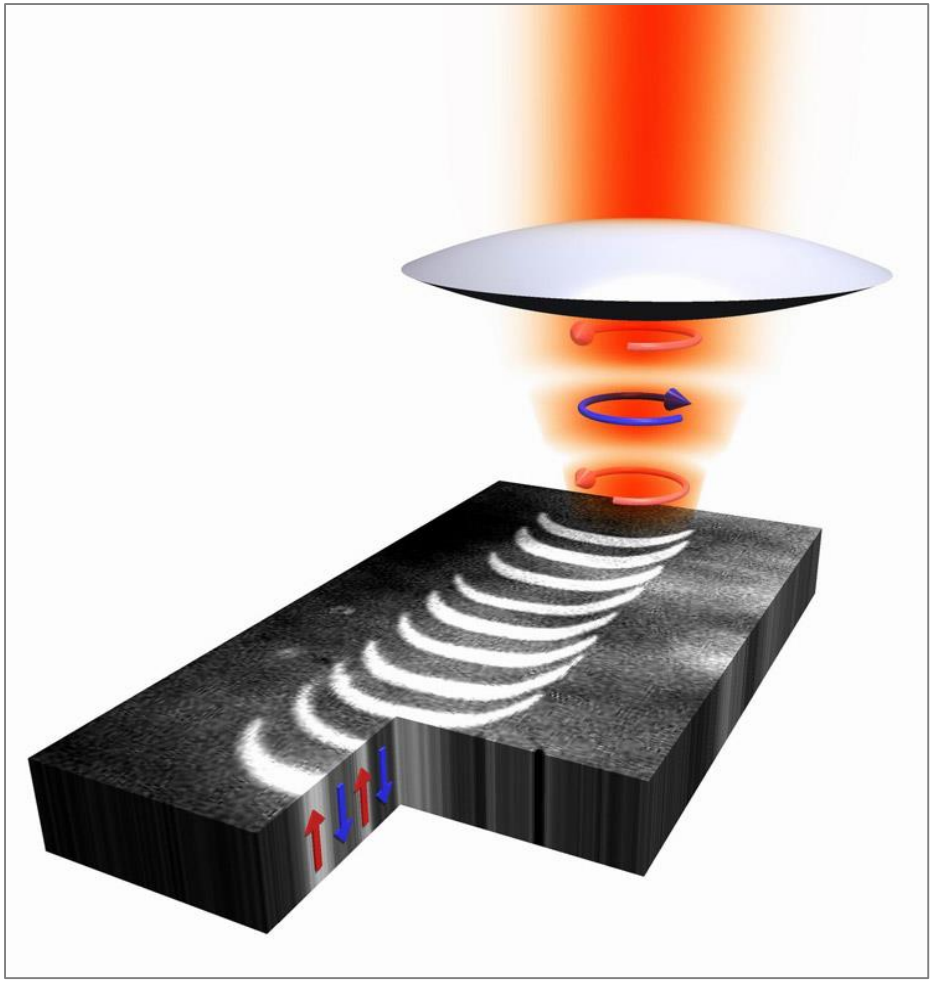
광학 기록

- 기록(쓰기) 과정


input
data
stream
11001001011101010101010
encoder/
modulator
current
drive
signal
laser
source
illumination
optics
newly
written
data
storage
medium
intense
light
beam
scan
spot

광학 기록

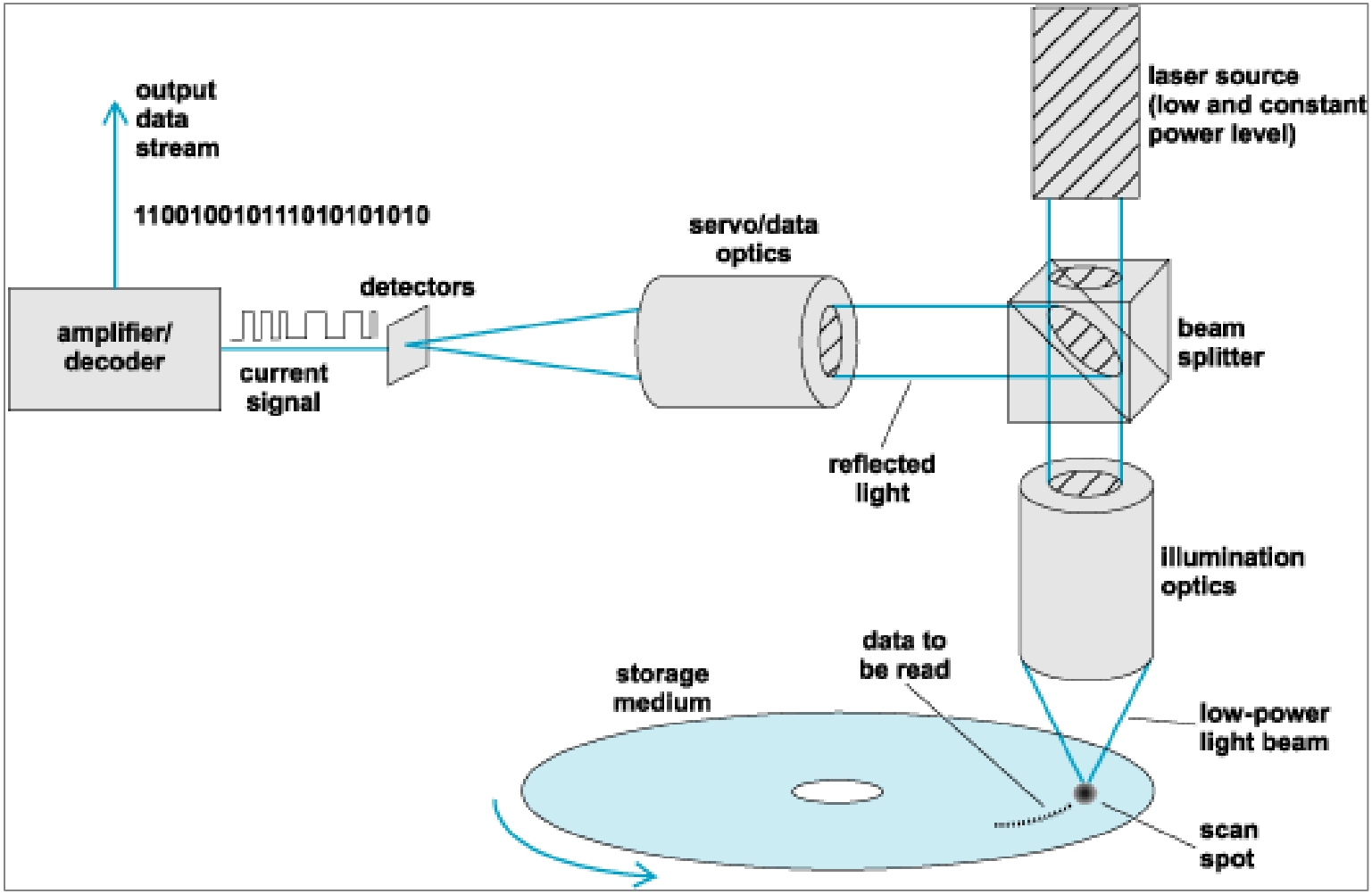
- 기록(쓰기) 과정



http://thefutureofthings.com/news/6186/laser-hard-drives-in-the-making.html

광학 기록

- 판독(읽기) 과정

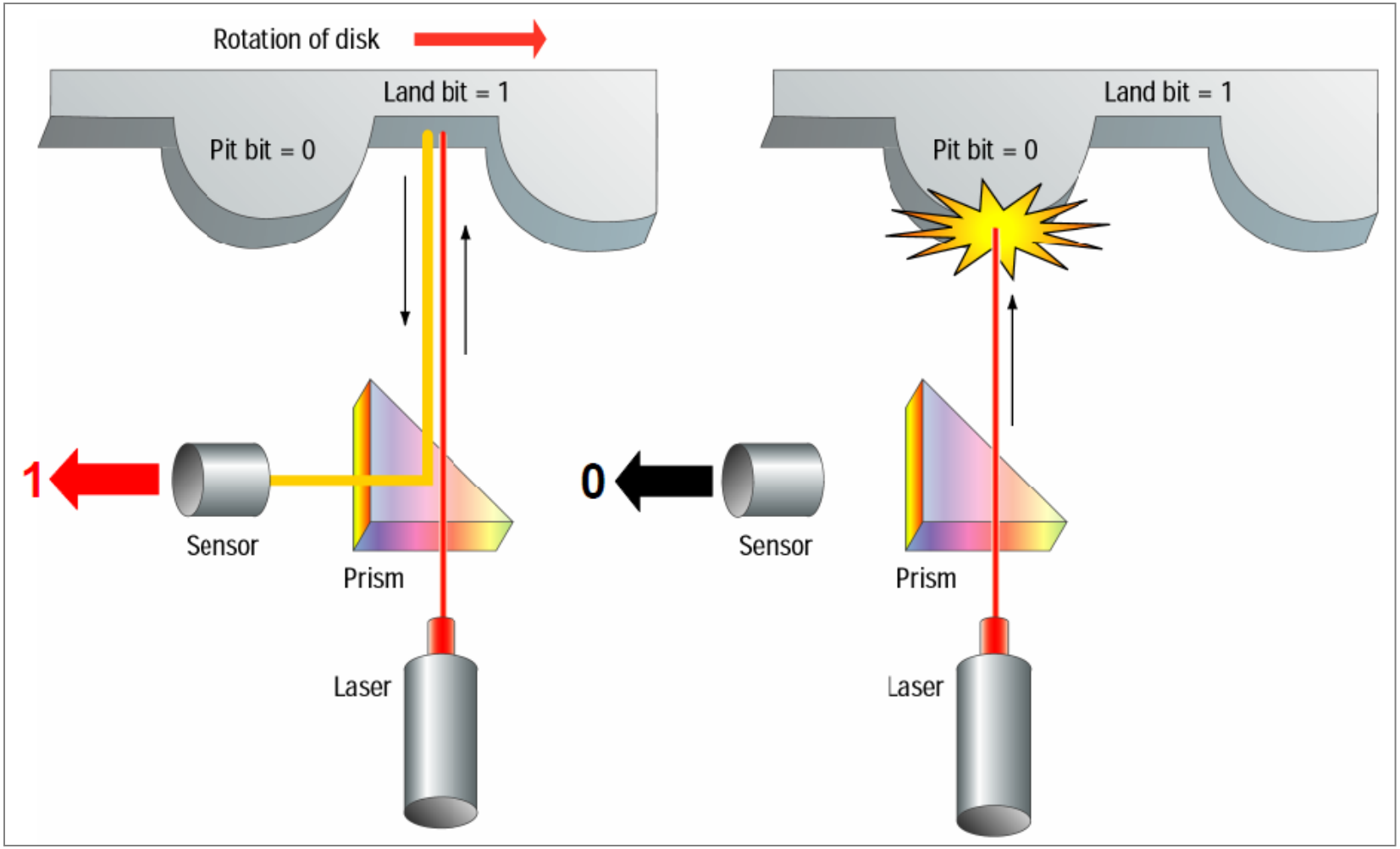


http://www.answers.com/topic/optical-recording

데이터 기록 방식

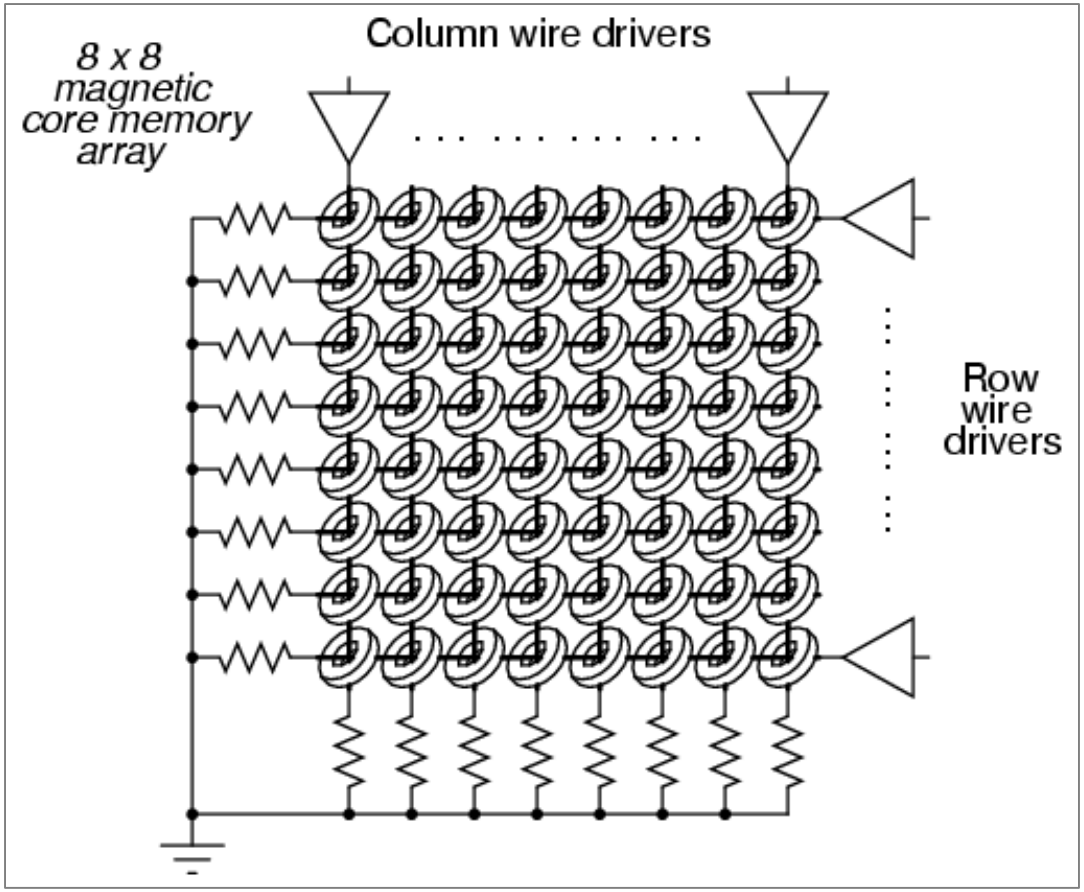
광학 기록

- 판독(읽기) 과정



전자 기록

- **트랜지스터 게이트**
  - 게이트 전압이 낮으면 1, 전압이 높으면 0

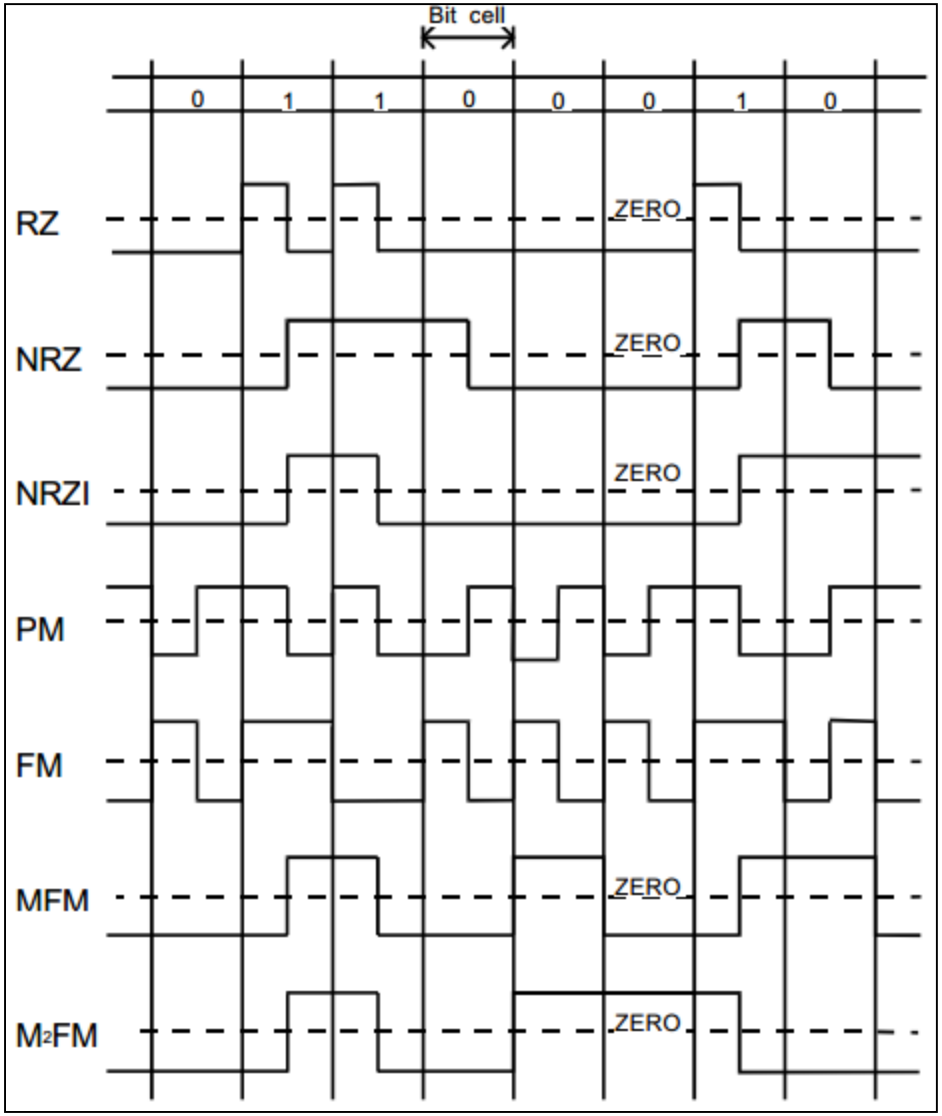


데이터 인코딩 방식

- 자기 인코딩 방식 (Magnetic Encoding)
- 광학 인코딩 방식 (Optical Encoding) ???
- 전자 인코딩 방식 (Electronic Encoding) ???

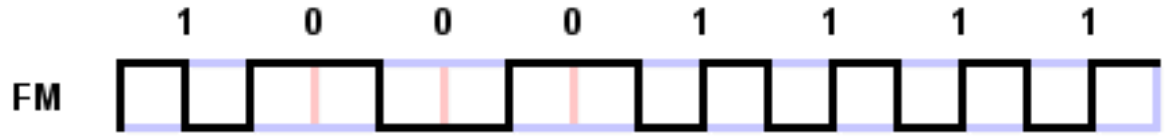
자기 인코딩 기법

- **인코딩 기법**
  - **RZ** (Return to Zero)
  - **NRZ** (Non-Return to Zero)
  - **NRZI** (Non-Return to Zero, Invert)
  - **PM** (Phase Modulation)
  - **FM** (Frequency Modulation)
  - **MFM** (Modified FM)
  - **MMFM** (Modified MFM)



자기 인코딩 기법

- **FM (Frequency Modulation); 주파수 변조** (0에 비해 1이 두 배의 반전 수를 가짐)
  - **0** : 자화 반전 다음에 자화 반전 X
  - **1** : 연속된 2개의 자화 반전
  - 1970년 말~1980년 초에 플로피 디스크에 사용 ➔ 후에 MFM으로 변경
  - 클록을 위해 추가된 자화 반전으로 인해 낭비가 매우 심함

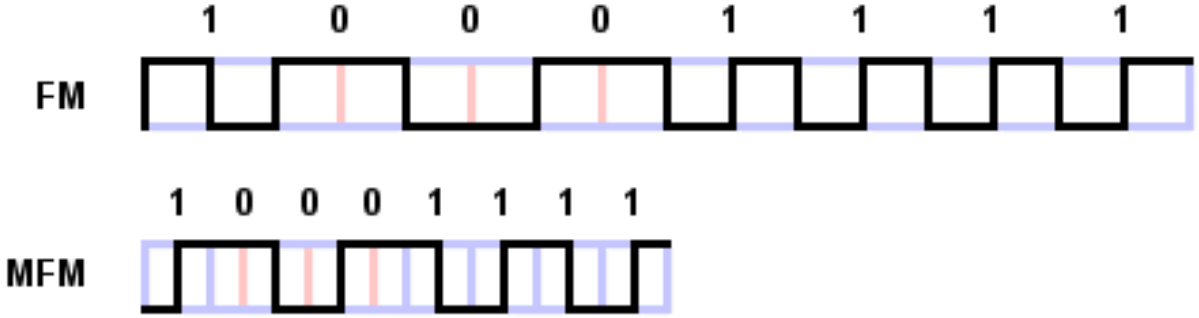


비트 패턴	인코딩 패턴	비트 당 자화 반전 수	임의 비트 패턴에서의 비율
0	RN	1	50%
1	RR	2	50%
가중 평균		1.5	100%

*** N** : No flux reversal, **R** : flux Reversal

자기 인코딩 기법

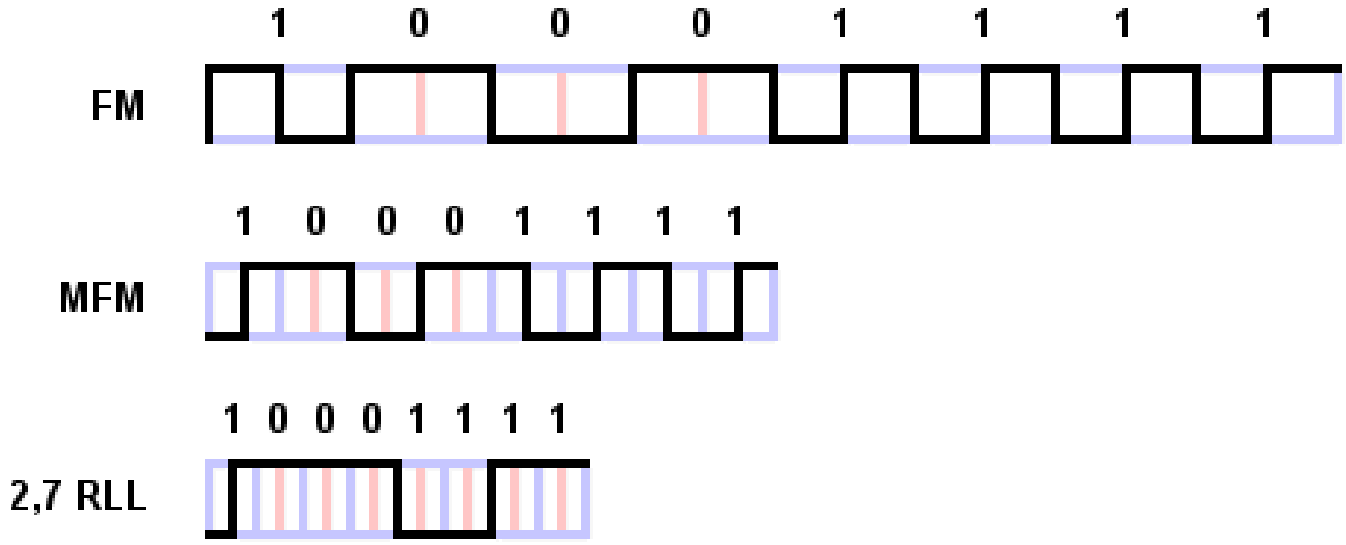
- **MFM (Modified Frequency Modulation); 변형 주파수 변조**
  - 클록에 사용되는 자화 반전 수를 줄여 FM을 향상 ➔ 연속된 0일 때만 자화 반전 사용
  - 알고리즘, 인코딩/디코딩 회로의 복잡 ➔ 어차피 컨트롤러가 하는 일이라
  - 플로피 디스크와 초기 하드디스크에 사용 ➔ 현재까지 플로피 디스크의 표준



비트 패턴	인코딩 패턴	비트 당 자화 반전 수	임의 비트 패턴에서의 비율
0 (0 다음에)	RN	1	25%
0 (1 다음에)	NN	0	25%
1	NR	1	50%
가중 평균		0.75	100%

자기 인코딩 기법

- **RLL (Run Length Limited); 런 길이 제한**
  - 하나의 비트를 인코딩 하는 것이 아닌 몇 비트를 묶어 패턴으로 기록 ➔ 클록/자화 반전 혼합
  - **런 길이(Run Length)** : 자화 반전 사이의 최소 간격
  - **런 제한(Run Limited)** : 자화 반전 사이의 최대 간격
  - 다양한 변형 존재 ➔ RLL (1,7), RLL (2,7)



자기 인코딩 기법

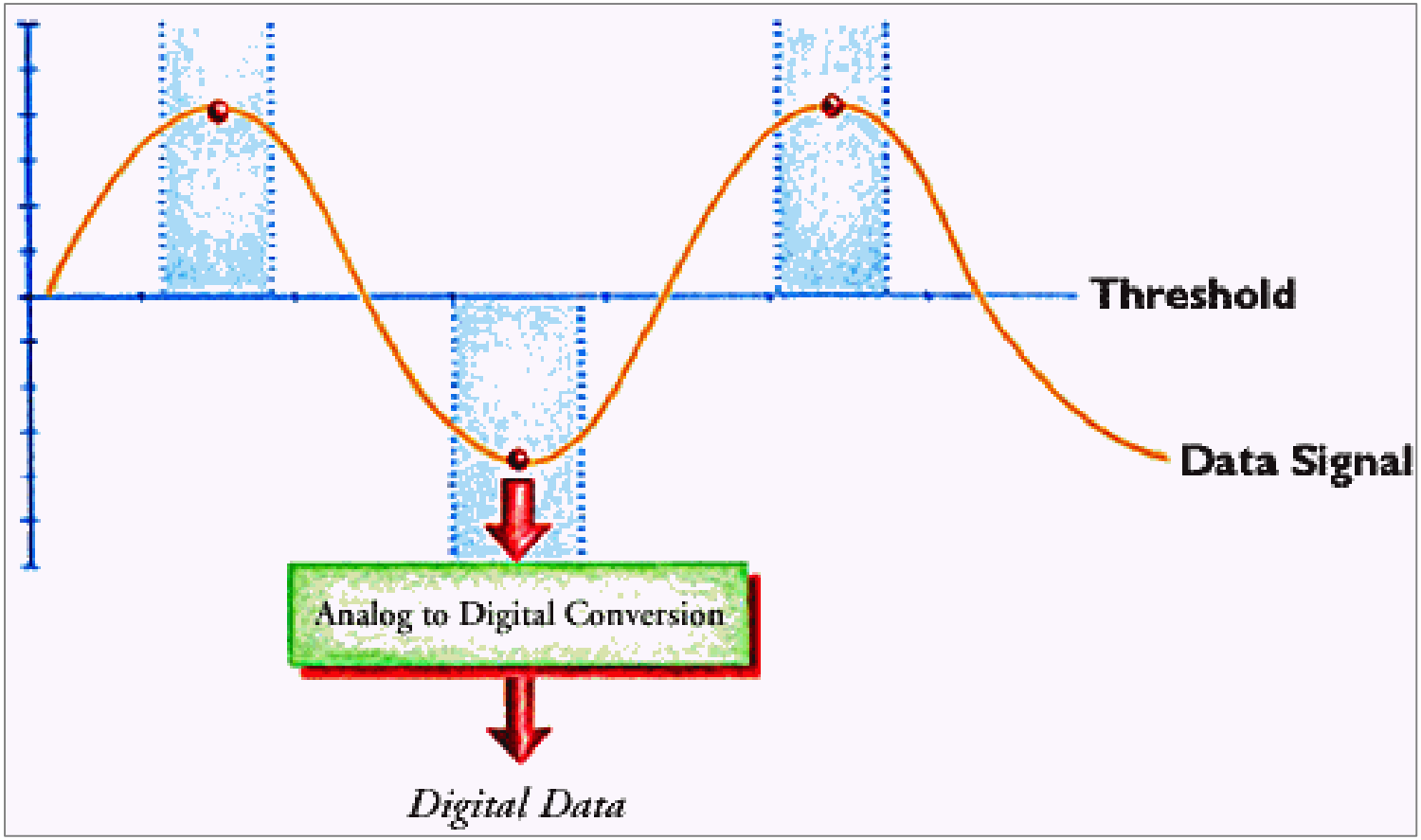
- **RLL (2, 7)**

비트 패턴	인코딩 패턴	비트 당 자화 반전 수	임의 비트 패턴에서의 비율
11	RNNN	1/2	25%
10	NRNN	1/2	25%
011	NNRNNN	1/3	12.5%
010	RNNRNN	2/3	12.5%
000	NNNRNN	1/3	12.5%
0010	NNRNNRNN	2/4	6.25%
0011	NNNNRNNN	1/3	6.25%
가중 평균		0.4635	100%

- RLL 등장으로 하드디스크 시장은 MFM에서 RLL로 교체
- 플로피 디스크는 여전히 MFM 사용

자기 인코딩 기법

- MFM, RLL 인코딩은 최대치 검출 (Peak Detection) 방식



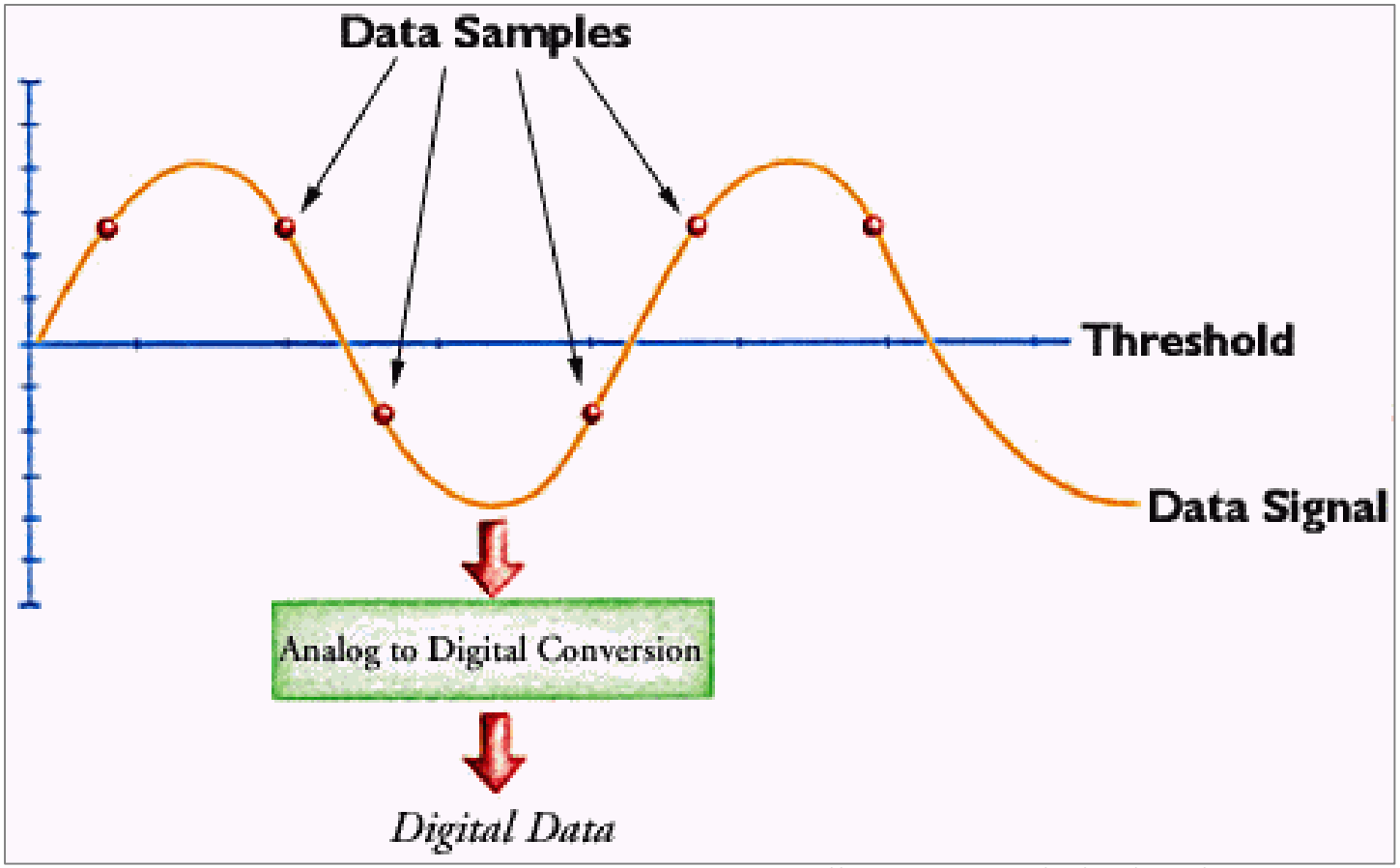
http://www.pcguide.com/ref/hdd/geom/dataPRML-c.html

자기 인코딩 기법

- **최대치 검출 방식의 한계**
  - 배경 잡음보다 최대치가 충분히 클 경우 잘 동작
  - 기록 밀도의 증가로 자화 반전의 최대치는 서로 더 밀접하게 위치 ➔ 간섭 발생
  - 간섭을 줄이고자 자기장 세기를 줄임 ➔ 최대치 검출의 어려움
  - 어려움을 해결하고자 등장한 것이 PRML (부분 응답, 최대 유사)
  - PRML은 RLL에 비해 30~40%의 기록밀도를 증가시킴

자기 인코딩 기법

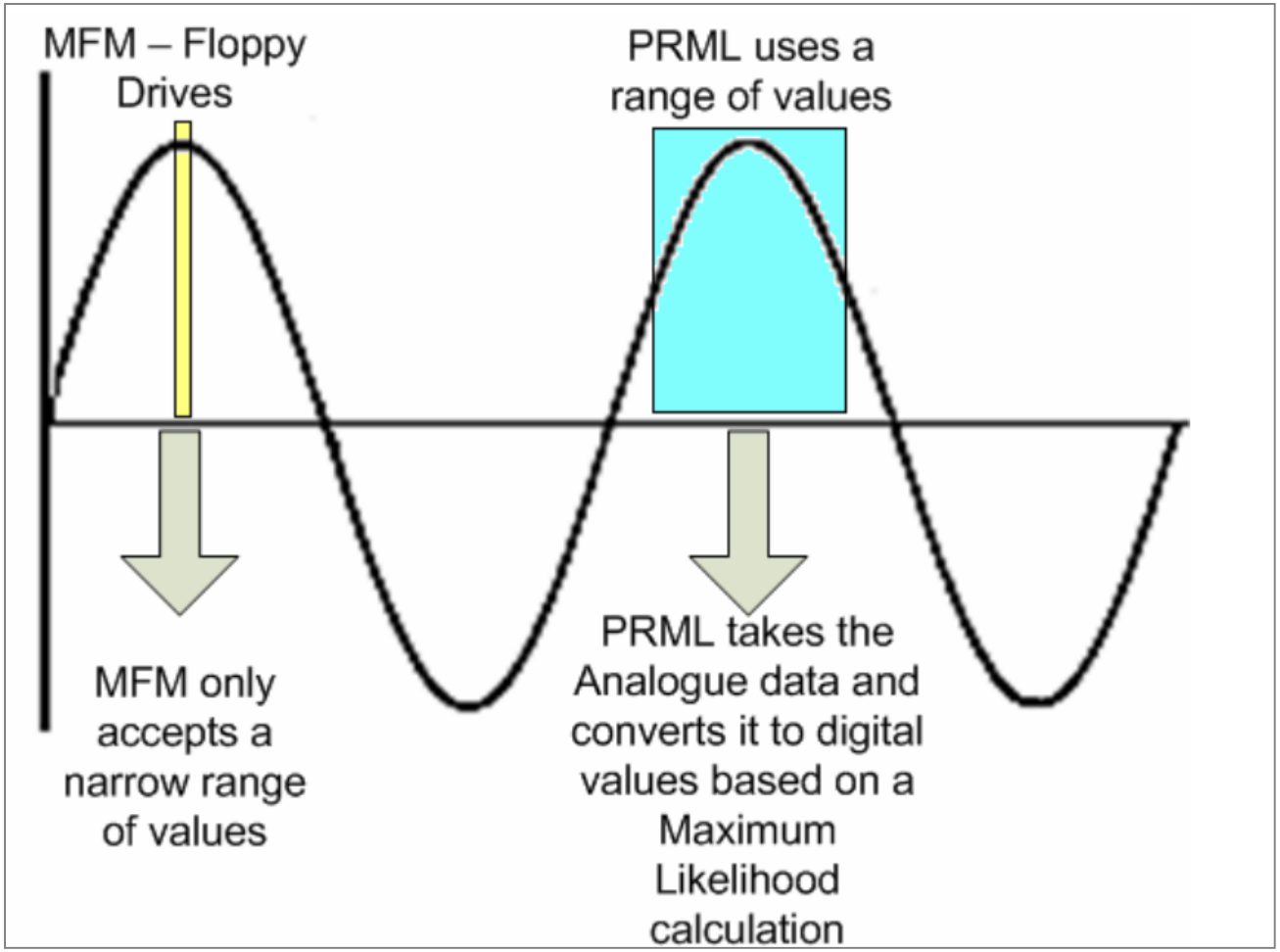
- **PRML (Partial Response, Maximum Likelihood); 부분 응답, 최대 유사**
  - 데이터 샘플링 (부분 응답) ➔ 가장 큰 시퀀스 탐지 (최대 유사)



http://www.pcguide.com/ref/hdd/geom/dataPRML-c.html

자기 인코딩 기법

- **MFM vs. PRML**



http://www.vidarholen.net/~vidar/overwriting_hard_drive_data.pdf

자기 인코딩 기법

- **EPRML (Extended PRML); 확장 PRML**
  - 알고리즘과 처리 회로 개선 ➔ 더 정확하게 해석
  - 오류 발생률 증가 없이 PRML에 비해 약 20%~70%까지 기록 밀도 증가
  - 현대의 하드디스크는 모두 EPRML 인코딩 방식 사용

물리적 손상 복구

물리적 손상 복구

물리적 손상의 특징

- 저장 매체의 물리적인 피해가 원인이 되어 발생
- 데이터 손실을 감소해야 함
- 대부분의 물리적인 피해는 개인 사용자가 복구하기 어려움

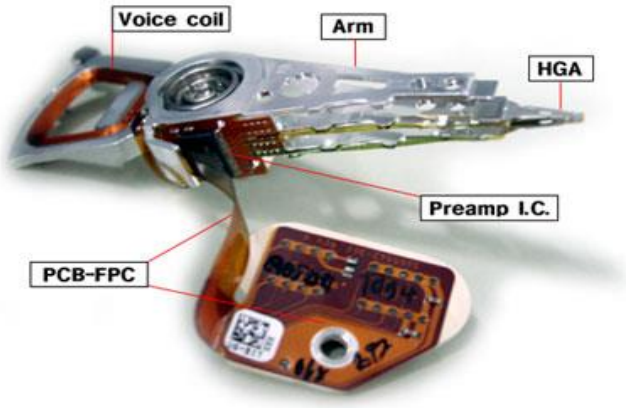
물리적 손상 복구

물리적 손상 유형과 복구율

- **디스크 충돌, 긁힘**
  - 디스크 크래시(Crash)으로 인해 마그네틱 부분의 손상
  - 헤드, 스핀들, 모터, 액추에이터 등이 함께 파손되는 복합적 손상이 많음
  - 평균 복구율 40%

- **논리보드 고장**
  - 논리보드의 고장으로 인해 정상적인 동작 불가
  - 헤드나 디스크 등과 함께 손상된 경우가 많음

HSA(Head Stack Assembly)



- **헤드 및 스핀들 모터 고장** **(돌다 말다 하는 경우)**
  - 보드 손상으로 모터가 구동하지 않을 경우 90% 복구율
  - 모터 자체가 손상된 경우 70% 이하 복구율
  - 모터 베이링 손상으로 축이 기울어진 경우 35% 복구율

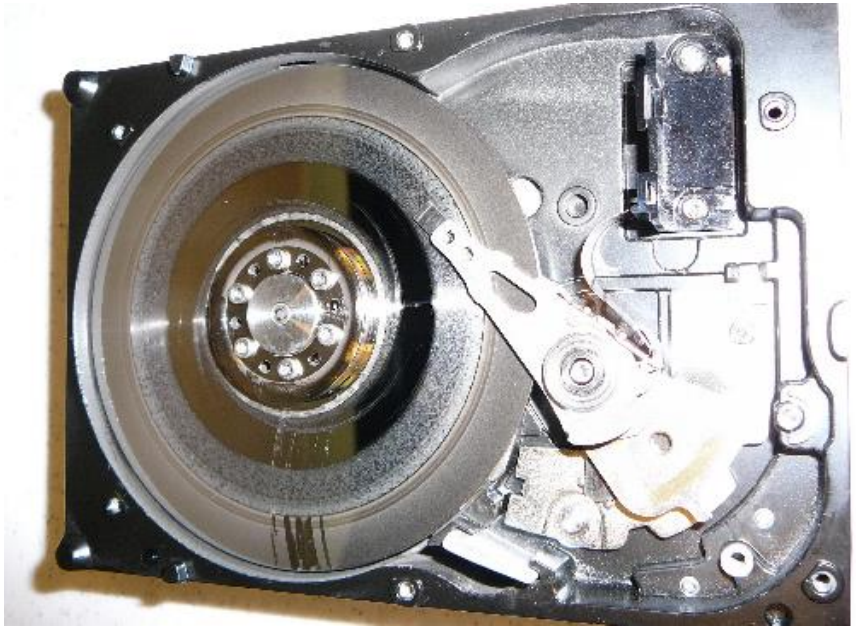
물리적 손상 복구

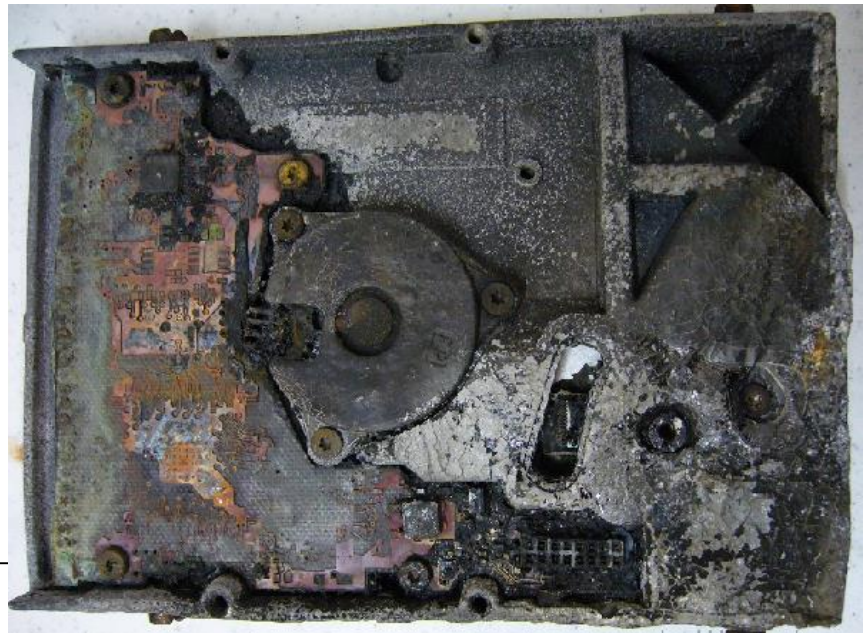
물리적 손상 유형과 복구율

- **물리적 불량 섹터**
  - 실린더 0이 손상된 경우 바이러스와 같은 증상
  - 불량 섹터가 발생한 파일이 열리지 않음
  - 복구율 70%

- **물리적인 충격**
  - 실수로 떨어트리거나 하여 충격을 준 경우
  - 스핀들 모터 축 이상, 헤드가 디스크 표면을 손상
  - 복구율 40%

- **자연재해(화재, 침수 등)**
  - 하드디스크 숨구멍을 통해 오염물 유입
  - 전원 공급 없이 완전히 마르기 전에 복구 시작





물리적 손상 복구 방법

- 대체 부품은 동일한 제조사 모델로 교체하는 것이 적절
- 원인 불명인 경우 논리보드 ➔ 헤드 ➔ 스핀들 모터 순으로 교환 및 복구 시도
- 먼지, 온도, 습도에 민감하므로 클린룸(Clean Room) 내에서 분해/조립 수행

http://www.drivecrash.com/clean-room/

논리적 손상 복구

논리적 손상 복구

논리적 손상 원인

- **논리적인 영역이 손상되어 발생**
  - 전원 공급의 비이상적인 차단(정전 등)
  - 하드웨어와 관련된 문제(특히, RAID 컨트롤러)
  - 시스템 크래시

- 별도의 장비를 사용하거나 논리적 구조의 변경을 통해 복구
- 저장매체의 논리적인 영역과 정보에 대한 이해가 필수적

논리적 손상 유형과 복구율

- **MBR(Master Boot Record) 손상**
  - 저장매체 LBA 0에 위치하는 512 바이트로 파티션 테이블과 부트 코드가 기록
  - 부트 코드가 손상된 경우, 동일 운영체제의 부트 코드와 교체하거나 부팅 가능한 파티션으로 점프
  - MBR 영역 손상 시 거의 100% 복구 가능

- **BR(Boot Record) 손상**
  - 파티션 정보 (BPB, BIOS Parameter Block)와 부트 코드(운영체제 로드)가 기록
  - 파티션 정보를 교정하고 동일 운영체제의 부트 코드로 교체
  - BR 영역 손상 시 거의 100% 복구 가능

- **파일시스템 메타데이터 손상, RAID 구조 손상**
  - FAT(File Allocation Table), 디렉터리 엔트리, MFT(Master File Table), RAID 메타데이터 손상
  - 상황에 따라 0 ~ 100% 복구 가능

논리적 손상 유형과 복구율

- **펌웨어 SA(Service Area) 손상**
  - SA 영역은 제조사에서 HDD 동작에 필요한 정보를 기록한 영역 ➔ 일반적으로 접근하기 어려움
  - 별도의 Factory Level의 장비(PC3000 등)를 이용해 접근 및 오류 검사
  - **복구 방안**
    - ✓ 손상이 발생한 데이터 수정이나 동일한 버전의 F/W로 덮어씀
    - ✓ 물리적으로 칩 교체

PC 3000의 주요 기능

- **Operation with HDD PCB**
  - 펌웨어 설정 값 확인, 자가진단 역할, Buffer RAM 테스트
  - 플래시 롬에서 펌웨어 데이터 설정 값을 읽고 쓰는 기능
  - 플래시 롬의 데이터 복사 기능

- **Operation with SA**
  - SA에서 배드섹터 테스트
  - SA 모듈 복구와 덮어쓰기 작업

- **Operation with HDD**
  - 표면 검사 및 배드 섹터 확인 : 로우 레벨(Low-Level) 포맷 절차에 의해 시행

- **Operation with Reading/Writing HDD Head**
  - 헤드 테스트

삭제된 파일 복구

삭제된 파일 복구

파일 삭제 유형

- **일반 삭제 ➔ 휴지통을 거치는 경우**
  - 휴지통 구조(INFO2, $I)에 흔적이 남음
  - 파일 메타데이터의 변경 ➔ 복원 가능성이 높음 (원본 메타데이터 + 휴지통 파일의 메타데이터)
- **바로 삭제 (SHIFT + DEL)**
  - 휴지통을 거치지 않는 경우
  - 원본의 흔적 이외에 별도의 흔적이 남지 않음
- **시스템 API를 이용한 삭제**
  - 바로 삭제와 동일하게 휴지통을 거치지 않고 삭제

삭제된 파일 복구 유형

- **파일시스템 메타 정보를 이용한 복구**
  - 파일 메타 정보가 존재하는 경우

- **데이터 카빙**
  - 파일 메타 정보가 덮어써진 경우

- **덮어써진 파일 복구**
  - 파일 데이터가 덮어써진 경우

파일시스템 메타 정보를 이용한 복구

- **파일시스템 추상적 구조**

- **각 파일시스템의 메타 정보**

  - **FAT12/16/32 :**

    ✓ 디렉터리 엔트리 정보 + FAT 영역

  - **exFAT**

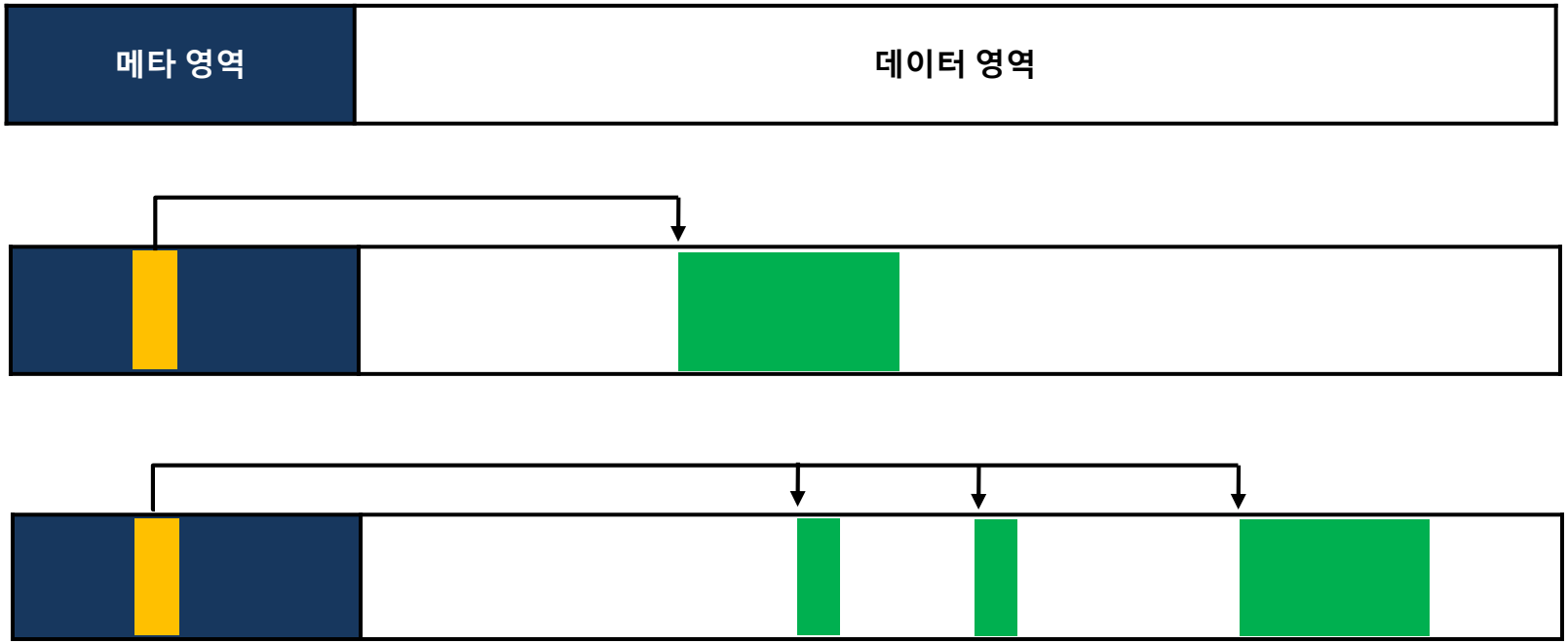
    ✓ 파일 디렉터리 + 스트림 확장(Stream Extension) + 파일 이름 확장(File Name Extension) 엔트리

  - **NTFS**

    ✓ MFT 엔트리 (STD_INFO, FNA, DATA 속성 등)

삭제된 파일 복구

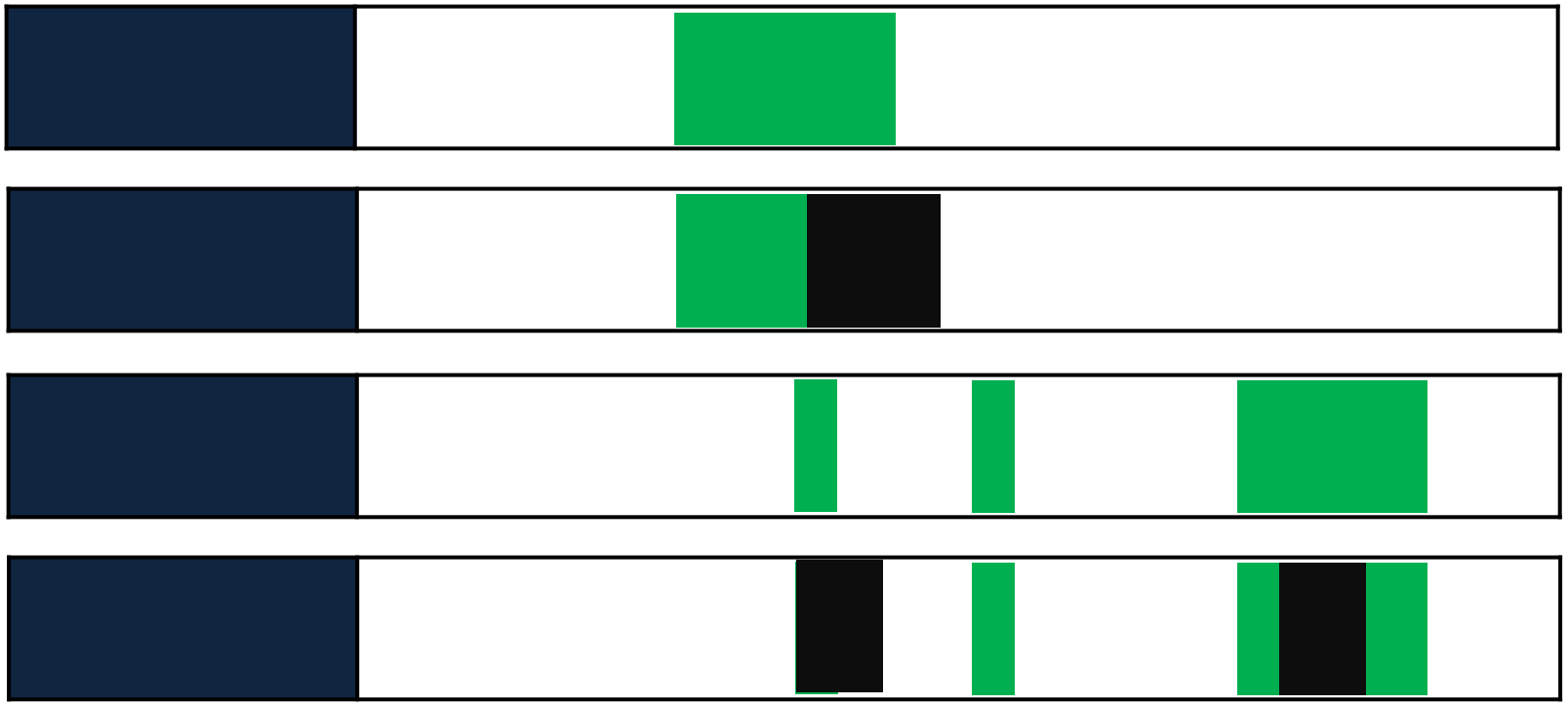
파일시스템 메타 정보를 이용한 복구



- 복구하고자 하는 파일의 메타 정보와 데이터가 온전하다면 100% 복구
- 운영체제 별로 차이가 있음
- 윈도우의 경우 시스템 파티션이냐 아니냐에 따라 차이가 있음

삭제된 파일 복구

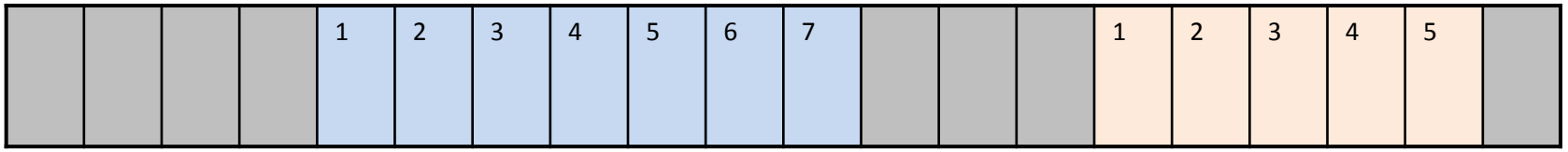
데이터 카빙



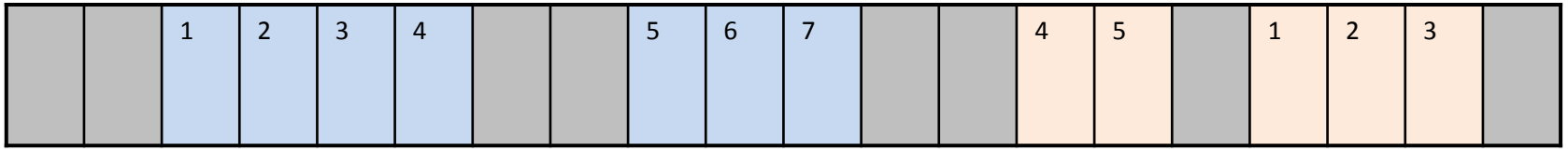
- 메타 정보가 없는 경우 개별 파일 구조에 기반하여 복구 ➔ 데이터 카빙 (Data Carving)
- 데이터 카빙은 바이너리 스트림에서 의미 있는 정보를 획득하는 기법
- 그래픽, 문서, 텍스트, 특정 데이터 구조, 문자열 등

데이터 카빙

- **연속적인 카빙**
  - 데이터가 저장매체의 연속된 공간에 저장된 경우 복구하는 기법



- **비연속적인 카빙**
  - 데이터의 단편화가 발생하여 저장매체의 여러 부분에 조각나 저장된 경우 복구하는 기법

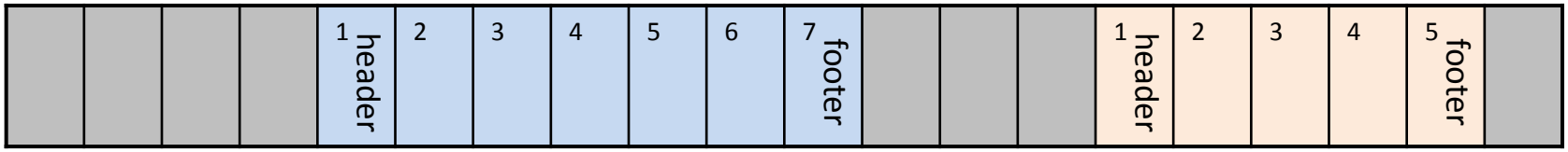


연속적인 카빙

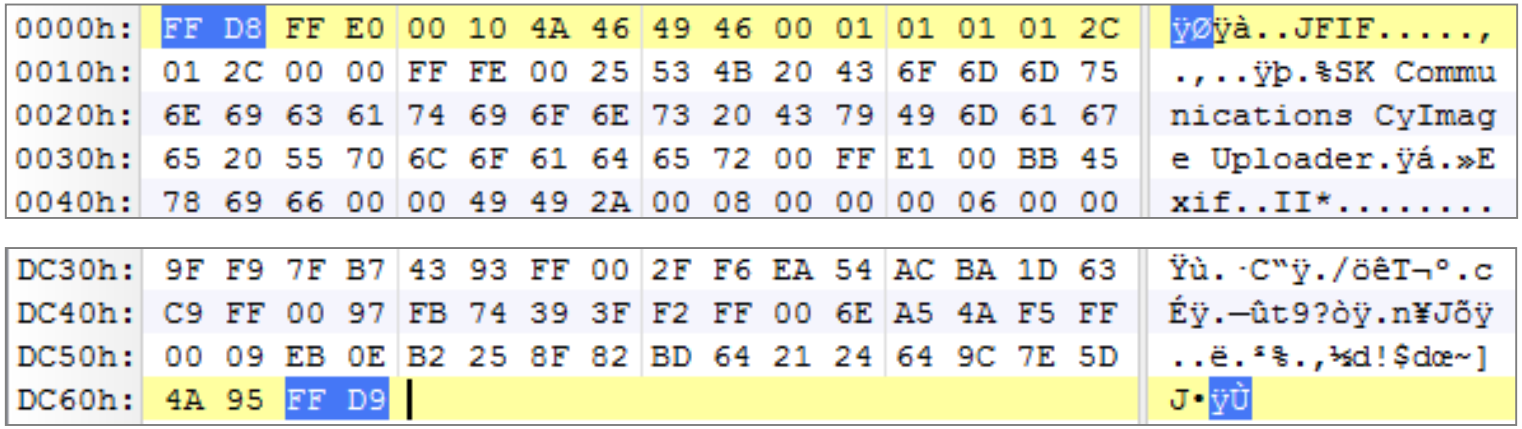
- **연속적인 카빙 기법**
  - 헤더/푸터 카빙
  - 램 슬랙 카빙
  - 파일 크기 카빙
  - 파일 검증 카빙

연속적인 카빙 기법

- **헤더/푸터 카빙**
  - 파일의 고유한 헤더와 푸터 시그니처를 이용한 카빙
  - 파일 시그니처 확인 : http://forensic-proof.com/archives/300



  - Ex) JPEG 파일의 헤더/푸터

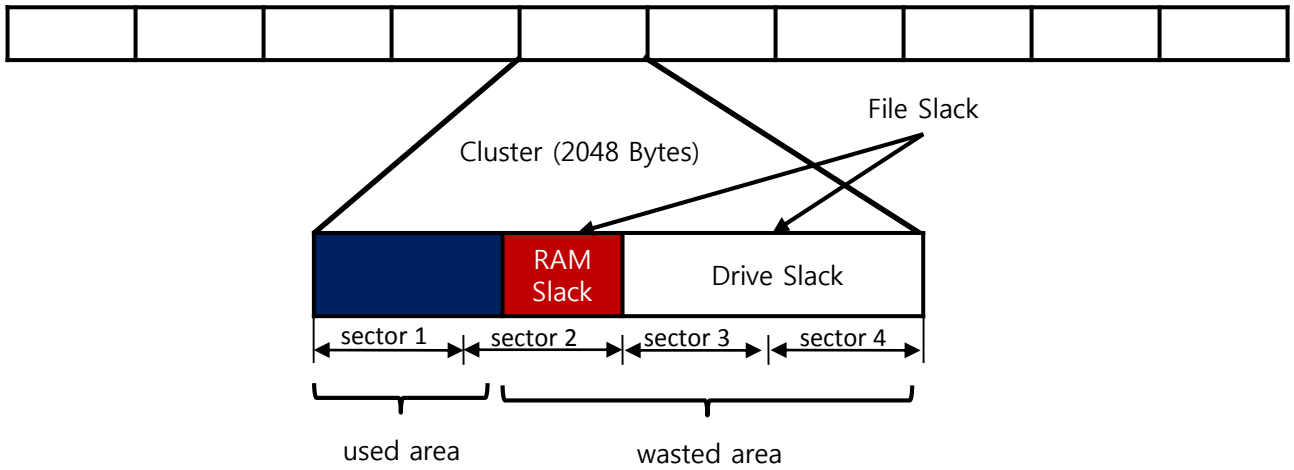


```
0000h: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 01 2C  ÿØÿà..JFIF.....,
0010h: 01 2C 00 00 FF FE 00 25 53 4B 20 43 6F 6D 6D 75  .,..ÿþ.%SK Commu
0020h: 6E 69 63 61 74 69 6F 6E 73 20 43 79 49 6D 61 67  nications CyImag
0030h: 65 20 55 70 6C 6F 61 64 65 72 00 FF E1 00 BB 45  e Uploader.ÿá.»E
0040h: 78 69 66 00 00 49 49 2A 00 08 00 00 00 06 00 00  xif..II*........

DC30h: 9F F9 7F B7 43 93 FF 00 2F F6 EA 54 AC BA 1D 63  Ÿù.·C"ÿ./öêT¬°.c
DC40h: C9 FF 00 97 FB 74 39 3F F2 FF 00 6E A5 4A F5 FF  Éÿ.—ût9?òÿ.n¥Jõÿ
DC50h: 00 09 EB 0E B2 25 8F 82 BD 64 21 24 64 9C 7E 5D  ..ë.²%.‚½d!$dœ~]
DC60h: 4A 95 FF D9                                      J•ÿÙ
```

연속적인 카빙 기법

- 램 슬랙 카빙



연속적인 카빙 기법

▪ 램 슬랙 카빙

- 윈도우 시스템(98 +)에서 램 슬랙이 항상 0x00으로 채워지는 성질을 이용한 카빙
  - ✓ 플래시 메모리에서는????
- 푸터가 존재하는 파일의 경우 단순한 헤더-푸터 카빙에 비해 오탐(False-Positive)이 줄어듦
- 푸터를 일일이 바이트 비교를 통해 찾아야 하므로 빠른 검색 알고리즘 사용
  - ✓ 예) Boyer-Moore Search
- 파일 형식별 평균 크기에 기반하여 최대 크기 제한을 설정해야 함

연속적인 카빙 기법

- **램 슬랙 카빙이 적용 가능한 파일 형식**

파일 형식	시그니처	
	헤더	푸터
JPEG	FF D8	FF D9
GIF	47 49 46 38 37 61 (“GIF87a”)	003B
	47 49 46 38 39 61 (“GIF89a”)	
PNG	89 50 4E 47 0D 0A 1A 0A	49 45 4E 44 AE 42 60 82
PDF	25 50 44 46 2D 31 2E ("PDF-1.")	25 25 45 4F 46 ("%%EOF")
HTML	"<HTML>"" 또는 “<html>"	"</HTML>" 또는 “</html>"
	"<!DOCTYPE HTML" 또는 "<!doctype html"	
PS	25 21 50 53 2D 41 64 6F 62 65 ("!PS-Adobe")	25 25 45 4F 46 ("%%EOF")
EPS	25 21 50 53 2D 41 64 6F 62 65 ("!PS-Adobe")	25 25 54 72 61 69 6C 65 72 xx ("%%Trailer")

연속적인 카빙 기법

- **파일 크기 카빙**
  - 파일 크기 카빙은 헤더만 존재하고 푸터 시그니처가 없는 파일에 효과적인 방법
  - 대부분의 파일은 파일의 시작 부분에 고유한 헤더 구조체를 가짐
  - 헤더 구조체 내용을 기반으로 파일 크기를 획득하거나 계산 가능
  - 램 슬랙 카빙에 비해 속도면에서 효과적
  - 헤더 데이터 구조만을 가지고 카빙을 수행하면 많은 오탐이 발생
  - 헤더 데이터 구조의 제한된 값을 검증

연속적인 카빙 기법

▪ 파일 크기 카빙

- Ex) BMP 파일 형식의 헤더 데이터 구조

바이트 범위	크기 (바이트)	설명
0 – 1	2	시그니처 (“BM”)
2 – 5	4	파일 크기
6 – 7	2	예약된 영역
8 – 9	2	예약된 영역
10 – 13	4	비트맵(bitmap) 데이터 시작 위치

바이트 범위	크기 (바이트)	설명
14 – 17	4	BMPINFOHEADER 크기 (항상 0x28)
18 – 21	4	이미지의 가로 크기
22 – 25	4	이미지의 세로 크기
26 – 27	2	색상 면의 수 (항상 0x0001)
28 – 29	2	픽셀 당 비트 수 (1, 4, 8, 16, 24, 32)
30 – 33	4	압축 방식 (0~5 사이의 값)
34 – 37	4	이미지의 바이트 크기
38 – 41	4	미터당 가로 픽셀 수
42 – 45	4	미터당 세로 픽셀 수
46 – 49	4	색상 테이블의 색상 중 실제 사용되는 색상 수
50 – 53	4	비트맵을 출력하는데 필수적인 색상 수

연속적인 카빙 기법

- **파일 크기 카빙이 적용 가능한 파일 형식**

파일 형식	헤더 시그니처 (16진수 값)
BMP	42 4D (“BM”)
TIFF	49 49 2A ("II*") 또는 4D 4D 2A ("MM*")
EXE	4D 5A (“MZ”)
DLL	
OBJ	
SYS	
ARJ	60 EA
ALZ	41 4C 5A 01
RAR	52 61 72 21 1A 07 ("Rar! ")
ZIP	50 4B 03 04 ("PK ")
DOCX	50 4B 03 04 ("PK ") + "[Content_Types].xml"
XLSX	
PPTX	

연속적인 카빙 기법

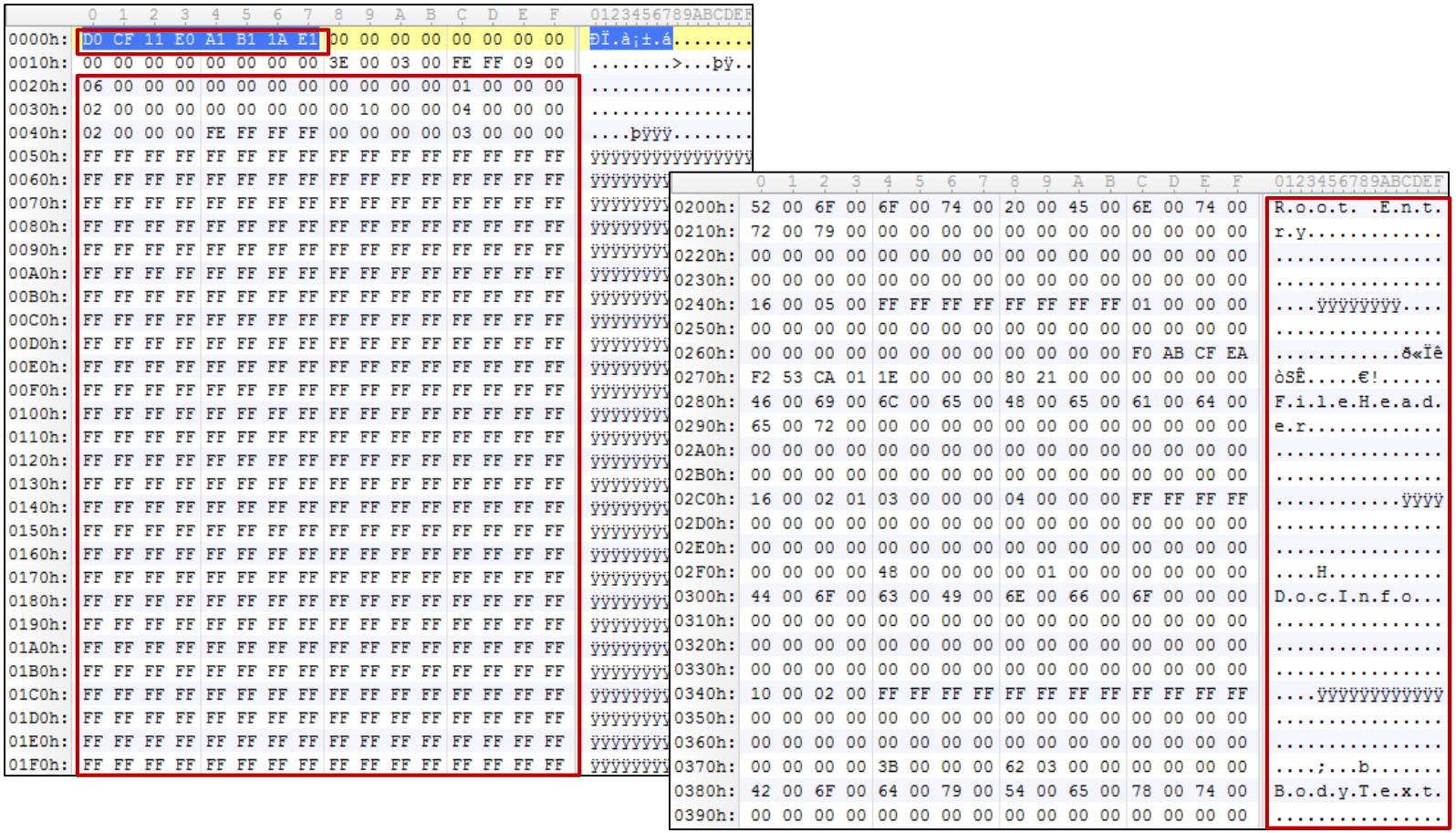
- **파일 검증 카빙**
  - 파일 크기만으로 파일의 유효성을 입증하기 어려움
  - 헤더 시그니처만 존재하고 파일 크기 정보를 얻기 어려운 파일 형식이 존재
  - 파일의 내부 구조 검증을 통해 응용프로그램 수준에서 확인이 가능한 선에서 복구
  - 내부 구조 검증에 소요되는 시간 vs 오탐 발생률

삭제된 파일 복구

연속적인 카빙 기법

- **파일 검증 카빙**

  - Ex) HWP 파일의 파일 검증



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	D0	CF	11	E0	A1	B1	1A	E1	00	00	00	00	00	00	00	00	ÐÏ.à¡±.á........
0010h:	00	00	00	00	00	00	00	00	3E	00	03	00	FE	FF	09	00	........>...þÿ..
0020h:	06	00	00	00	00	00	00	00	00	00	00	00	01	00	00	00	....
0030h:	02	00	00	00	00	00	00	00	00	10	00	00	04	00	00	00	....
0040h:	02	00	00	00	FE	FF	FF	FF	00	00	00	00	03	00	00	00	....þÿÿÿ........
0050h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0060h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0070h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0080h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0090h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00A0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00B0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00C0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00D0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00E0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00F0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0100h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0110h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0120h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0130h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0140h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0150h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0160h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0170h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0180h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0190h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01A0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01B0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01C0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01D0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01E0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
01F0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0200h:	52	00	6F	00	6F	00	74	00	20	00	45	00	6E	00	74	00	R.o.o.t. .E.n.t.
0210h:	72	00	79	00	00	00	00	00	00	00	00	00	00	00	00	00	r.y.............
0220h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0230h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0240h:	16	00	05	00	FF	FF	FF	FF	FF	FF	FF	FF	01	00	00	00	....ÿÿÿÿÿÿÿÿ....
0250h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0260h:	00	00	00	00	00	00	00	00	00	00	00	00	F0	AB	CF	EA	............ð«Ïê
0270h:	F2	53	CA	01	1E	00	00	00	80	21	00	00	00	00	00	00	òSÊ.....€!......
0280h:	46	00	69	00	6C	00	65	00	48	00	65	00	61	00	64	00	F.i.l.e.H.e.a.d.
0290h:	65	00	72	00	00	00	00	00	00	00	00	00	00	00	00	00	e.r.............
02A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
02B0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
02C0h:	16	00	02	01	03	00	00	00	04	00	00	00	FF	FF	FF	FF	............ÿÿÿÿ
02D0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
02E0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
02F0h:	00	00	00	00	48	00	00	00	00	01	00	00	00	00	00	00	....H...........
0300h:	44	00	6F	00	63	00	49	00	6E	00	66	00	6F	00	00	00	D.o.c.I.n.f.o...
0310h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0320h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0330h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0340h:	10	00	02	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	....ÿÿÿÿÿÿÿÿÿÿÿÿ
0350h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0360h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0370h:	00	00	00	00	3B	00	00	00	62	03	00	00	00	00	00	00	....;...b.......
0380h:	42	00	6F	00	64	00	79	00	54	00	65	00	78	00	74	00	B.o.d.y.T.e.x.t.
0390h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....

연속적인 카빙 기법

▪ 파일 검증 카빙

- Ex) HWP 파일의 스토리지와 스트림

스토리지와 스트림	설명
FileHeader	파일 메타 정보
DocInfo	문서 정보
BodyText – Section N	본문 정보
PrvText	미리보기 텍스트
PrvImage	미리보기 이미지
DefaultJScript	스크립트 매크로 정보
JScriptVersion	스크립트 버전 정보
DocOptions - _LinkDoc	연결 문서 정보
BinData – BIN *.*	이미지 파일 정보
HwpSummary Information	문서 요약 정보

비연속적인 카빙

- **비연속적인 카빙 기법**
  - 파일 조각을 찾는 것이 관건 ➔ 파일이 조각이 나긴 하나? (http://forensic-proof.com/archives/320)
  - 파일 형식에 따른 데이터 특성을 이용함
    - ✓ 시그니처 (Signature) 탐색
    - ✓ 엔트로피 (Entropy)
    - ✓ 바이트 분포 (Byte Frequency Distribution)
    - ✓ 바이트 편차 (Rate of Change)
    - ✓ ASCII 빈도 (Frequency)
    - ✓ 공통된 패턴(긴 문자열) 비교

비연속적인 카빙

- 시그니처에 기반한 기법

  - Ex) JPEG

축약어	이름	시그니처
SOI	Start Of Image	**0xFF, 0xD8**
SOF0	Start Of Frame (Baseline DCT)	**0xFF, 0xC0**
SOF2	Start Of Frame (Progressive DCT)	**0xFF, 0xC2**
DHT	Define Huffman Table(s)	**0xFF, 0xC4**
DQT	Define Quantization Table(s)	**0xFF, 0xDB**
DRI	Define Restart Interval	**0xFF, 0xDD**
SOS	Start Of Scan	**0xFF, 0xDA**
RST*n*	Restart	**0xFF, 0xD0 ... 0xD7**
APP*n*	Application-specific	**0xFF, 0xE*n***
COM	Comment	**0xFF, 0xFE**
EOI	End Of Image	**0xFF, 0xD9**

✓ 모든 포맷에 적용하기 어려움

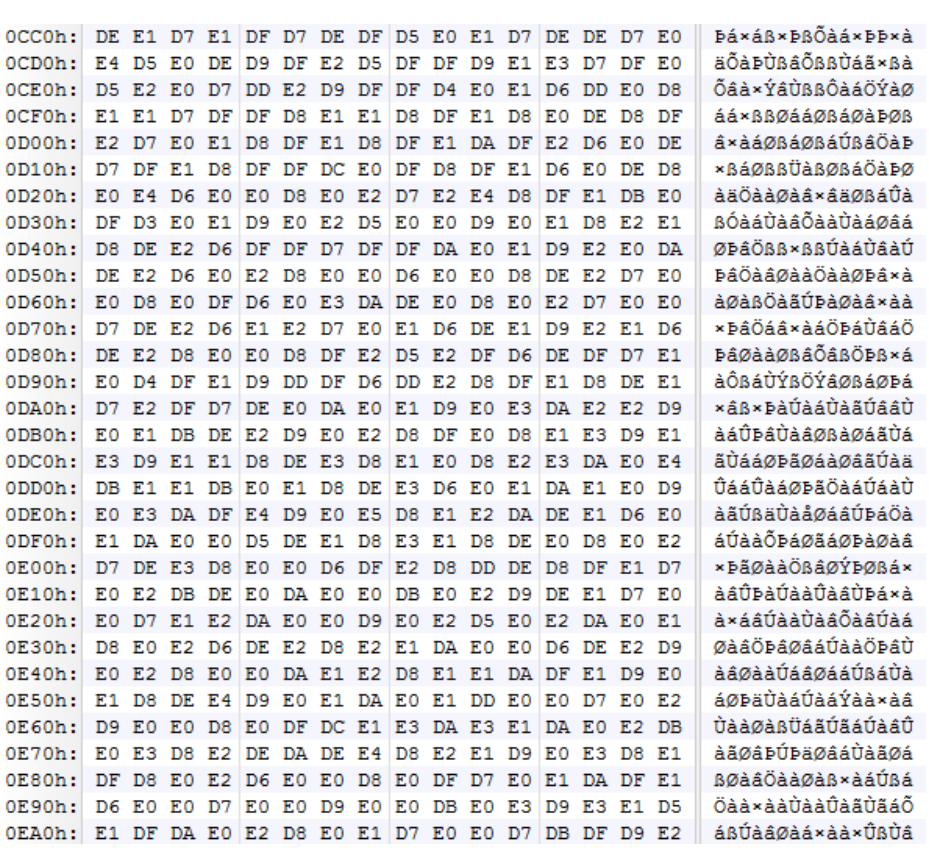
✓ 오탐(false-positive)가 많이 발생함

삭제된 파일 복구

비연속적인 카빙

$$H(W) = -\sum_{c \in S_W} P(c)\log_2(P(c))$$

- **엔트로피(Entropy)를 이용한 기법**
  - 블록/클러스터의 엔트로피를 계산하여 특정 파일의 조각을 분류



0CC0h:	DE	E1	D7	E1	DF	D7	DE	DF	D5	E0	E1	D7	DE	DE	D7	E0	Þá×á ß×ÞßÕàá×ÞÞ×à
0CD0h:	E4	D5	E0	DE	D9	DF	E2	D5	DF	DF	D9	E1	E3	D7	DF	E0	äÕàÞÙßâÕßßÙáã×ßà
0CE0h:	D5	E2	E0	D7	DD	E2	D9	DF	DF	D4	E0	E1	D6	DD	E0	D8	Õâà×ÝáÙßßÔàáÖÝàØ
0CF0h:	E1	E1	D7	DF	DF	D8	E1	E1	D8	DF	E1	D8	E0	DE	D8	DF	áá×ßßØááØßáØàÞØß
0D00h:	E2	D7	E0	E1	D8	DF	E1	D8	DF	E1	DA	DF	E2	D6	E0	DE	â×àáØßáØßáÚßâÖàÞ
0D10h:	D7	DF	E1	D8	DF	DF	DC	E0	DF	D8	DF	E1	D6	E0	DE	D8	×ßáØßßÜàßØßáÖàÞØ
0D20h:	E0	E4	D6	E0	E0	D8	E0	E2	D7	E2	E4	D8	DF	E1	DB	E0	àäÖààØàâ×âäØßáÛà
0D30h:	DF	D3	E0	E1	D9	E0	E2	D5	E0	E0	D9	E0	E1	D8	E2	E1	ßÓàáÙàâÕààÙàáØâá
0D40h:	D8	DE	E2	D6	DF	DF	D7	DF	DF	DA	E0	E1	D9	E2	E0	DA	ØÞâÖßß×ßßÚàáÙâàÚ
0D50h:	DE	E2	D6	E0	E2	D8	E0	E0	D6	E0	E0	D8	DE	E2	D7	E0	ÞâÖàâØààÖààØÞâ×à
0D60h:	E0	D8	E0	DF	D6	E0	E3	DA	DE	E0	D8	E0	E2	D7	E0	E0	àØàßÖàãÚÞàØàâ×àà
0D70h:	D7	DE	E2	D6	E1	E2	D7	E0	E1	D6	DE	E1	D9	E2	E1	D6	×ÞâÖáâ×àáÖÞáÙâáÖ
0D80h:	DE	E2	D8	E0	E0	D8	DF	E2	D5	E2	DF	D6	DE	DF	D7	E1	ÞâØààØßâÕâßÖÞß×á
0D90h:	E0	D4	DF	E1	D9	DD	DF	D6	DD	E2	D8	DF	E1	D8	DE	E1	àÔßáÙÝßÖÝâØßáØÞá
0DA0h:	D7	E2	DF	D7	DE	E0	DA	E0	E1	D9	E0	E3	DA	E2	E2	D9	×âß×ÞàÚàáÙàãÚââÙ
0DB0h:	E0	E1	DB	DE	E2	D9	E0	E2	D8	DF	E0	D8	E1	E3	D9	E1	àáÛÞâÙàáØßàØáãÙá
0DC0h:	E3	D9	E1	E1	D8	DE	E3	D8	E1	E0	D8	E2	E3	DA	E0	E4	ãÙááØÞãØáàØâãÚàä
0DD0h:	DB	E1	E1	DB	E0	E1	D8	DE	E3	D6	E0	E1	DA	E1	E0	D9	ÛááÛàáØÞãÖàáÚáàÙ
0DE0h:	E0	E3	DA	DF	E4	D9	E0	E5	D8	E1	E2	DA	DE	E1	D6	E0	àãÚßäÙàåØááÚÞáÖà
0DF0h:	E1	DA	E0	E0	D5	DE	E1	D8	E3	E1	D8	DE	E0	D8	E0	E2	áÚààÕÞáØãáØÞàØàâ
0E00h:	D7	DE	E3	D8	E0	E0	D6	DF	E2	D8	DD	DE	D8	DF	E1	D7	×ÞãØààÖßáØÝÞØß×
0E10h:	E0	E2	DB	DE	E0	DA	E0	E0	DB	E0	E2	D9	DE	E1	D7	E0	àâÛÞàÚààÛàâÙÞá×à
0E20h:	E0	D7	E1	E2	DA	E0	E0	D9	E0	E2	D5	E0	E2	DA	E0	E1	à×áâÚààÙàáÕàâÚàá
0E30h:	D8	E0	E2	D6	DE	E2	D8	E2	E1	DA	E0	E0	D6	DE	E2	D9	ØàâÖÞâØáâÚààÖÞâÙ
0E40h:	E0	E2	D8	E0	E0	DA	E1	E2	D8	E1	E1	DA	DF	E1	D9	E0	àâØààÚáâØááÚßáÚà
0E50h:	E1	D8	DE	E4	D9	E0	E1	DA	E0	E1	DD	E0	E0	D7	E0	E2	áØÞäÙàáÚàáÝàà×ââ
0E60h:	D9	E0	E0	D8	E0	DF	DC	E1	E3	DA	E3	E1	DA	E0	E2	DB	Ùàà Øà ßÙáßÛßáÚàáÛ
0E70h:	E0	E3	D8	E2	DE	DA	DE	E4	D8	E2	E1	D9	E0	E3	D8	E1	àãØâÞÚÞäØááÙàãØá
0E80h:	DF	D8	E0	E2	D6	E0	E0	D8	E0	DF	D7	E0	E1	DA	DF	E1	ßØàâÖààØàß×àáÚßá
0E90h:	D6	E0	E0	D7	E0	E0	D9	E0	E0	DB	E0	E3	D9	E3	E1	D5	Öàà×ààÙààÛàãÙãáÕ
0EA0h:	E1	DF	DA	E0	E2	D8	E0	E1	D7	E0	E0	D7	DB	DF	D9	E2	áßÚàâØàà×àà×ÛßÙâ

BMP

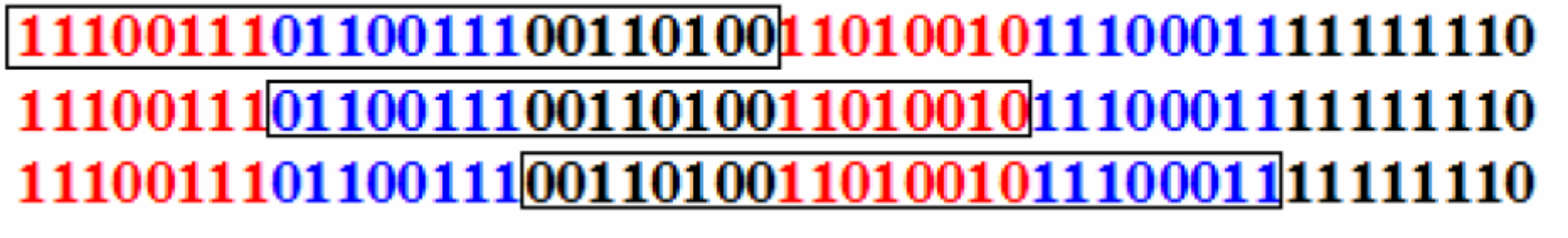
0CE0h:	6A	A5	3D	3D	74	6C	66	03	9B	20	D8	5F	B5	9F	4D	10	j¥==tlf.› Ø_µŸM.
0CF0h:	F9	F0	B9	EF	58	73	7B	F2	E8	E1	C7	61	0A	55	4F	ED	ùð¹ïXs{òèáÇa.UOí
0D00h:	F5	5F	65	DF	F9	D1	E6	5C	ED	CC	EF	1E	B4	E5	58	C6	õ_eßùÑæ\íÌï.´åXÆ
0D10h:	87	30	5F	CF	A6	AF	94	C2	5D	D6	1F	0A	A8	AD	CC	98	‡0_Ï¦¯"Â]Ö..¨-Ì˜
0D20h:	A1	B7	3F	FA	A4	BD	9C	4D	FA	9D	4E	D0	42	74	7F	7D	¡·?ú¤½œMú.NÐBt.}
0D30h:	B2	78	18	D9	34	B0	D4	35	F9	6D	EE	68	F1	4E	E6	71	²x.Ù4°Ô5ùmîhñNæq
0D40h:	FA	D2	26	36	E1	70	58	FE	FB	A3	35	53	0E	2F	BC	77	úÒ&6ápXþû£5S./¼w
0D50h:	53	E2	82	1B	16	4D	CF	F6	70	92	AF	94	1C	D4	6E	34	Sâ‚..MÏöp'¯"..Ôn4
0D60h:	A3	58	14	2F	FA	41	1C	1C	AE	F5	BE	17	60	CF	FA	A8	£X./úA..®õ¾.`Ïú¨
0D70h:	2C	4C	EC	B1	0E	9A	4A	AE	65	CD	FE	30	58	75	9D	38	,Lì±.šJ®eÍþ0Xu.8
0D80h:	D9	6A	CE	77	EA	DB	A6	B3	6C	82	5F	F5	CC	A1	5E	9A	ÙjÎwêÛ¦³l‚_õÌ¡^š
0D90h:	EF	7E	DE	90	D7	83	B9	11	C1	9B	A7	36	4F	AF	38	B3	ï~Þ.×ƒ¹.Á›§6O¯8³
0DA0h:	7F	70	7F	ED	A7	2F	5B	5F	DF	49	9E	DC	23	3D	ED	33	.p.í§/[_ßIžÜ#=í3
0DB0h:	E9	70	37	80	38	76	E9	B5	DF	27	BB	83	9D	C7	C9	3F	ép7€8véµß'»ƒ.ÇÉ?
0DC0h:	CA	76	13	34	3E	55	25	AC	69	1A	92	1A	59	1F	3B	C8	Êv.4>U%¬i.'.Y.;È
0DD0h:	92	32	EE	A6	3F	AC	15	7A	BA	51	BB	52	EC	4E	F7	AF	'2î¦?¬.z°Q»RìN÷¯
0DE0h:	41	C7	96	BB	87	97	F6	96	CC	7A	96	95	34	D3	B7	46	AÇ–»‡—ö–Ìz–•4Ó·F
0DF0h:	E3	61	C8	43	16	D1	7E	20	AC	75	FD	28	E5	CD	A3	59	ãaÈC.Ñ~ ¬uý(åÍ£Y
0E00h:	CE	74	FD	94	CA	FE	26	81	99	B1	67	B7	1E	54	BC	1F	Îtý"Êþ&.™±g·.T¼.
0E10h:	E9	4D	97	77	F8	15	F6	9C	DC	3F	A3	F6	75	70	62	D1	éM—wø.öœÜ?£öupbÑ
0E20h:	9D	F2	29	05	36	FB	EB	1F	F9	90	2D	E7	7C	3F	FF	9A	.ò).6ûë.ù.–ç|?ÿš
0E30h:	43	33	9B	D0	D4	6B	DF	6F	32	75	EC	46	F1	C5	06	97	C3›ÐÔkßo2uìFñÅ.—
0E40h:	4D	43	51	33	5D	A3	F9	03	E3	C9	EF	4B	C8	03	E4	42	MCQ3]£ù.ãÉïKÈ.äB
0E50h:	BD	90	8E	9A	B2	BD	77	B2	D2	C6	B4	37	7E	BB	A0	2E	½.ŽšªwªÒÆ´7~» .
0E60h:	3A	65	7C	ED	DD	B2	C8	ED	3F	E2	F6	79	8F	A7	4E	F8	:e|íÝ²Èí?âöy.§Nø
0E70h:	8C	14	9E	2F	29	FD	C4	2A	1E	95	A4	E7	98	29	76	A8	Œ.ž/)ýÄ*.•¤ç˜)v¨
0E80h:	4E	37	14	CA	DD	0D	BA	A1	76	E3	A8	BD	83	03	9B	90	N7.ÊÝ.°¡vã¨½ƒ.›.
0E90h:	EC	CB	26	14	5D	9F	98	DB	A3	11	F6	2E	79	3C	76	34	ìË&.]Ÿ˜Û£.ö.y<v4
0EA0h:	5F	DB	79	B0	8F	4D	F8	75	79	33	33	EE	B8	6C	10	79	_Ûy°.Møuy33î¸l.y
0EB0h:	62	46	CD	B8	A0	27	6C	48	88	62	F1	AE	F0	45	D3	73	bFÍ¸ 'lHˆbñ®ðEÓs
0EC0h:	36	E1	63	2A	CB	7C	6F	15	DD	E9	19	AD	B2	EA	44	80	6ác*Ë|o.Ýé.-²êD€

ZIP(Compressed)

삭제된 파일 복구

비연속적인 카빙

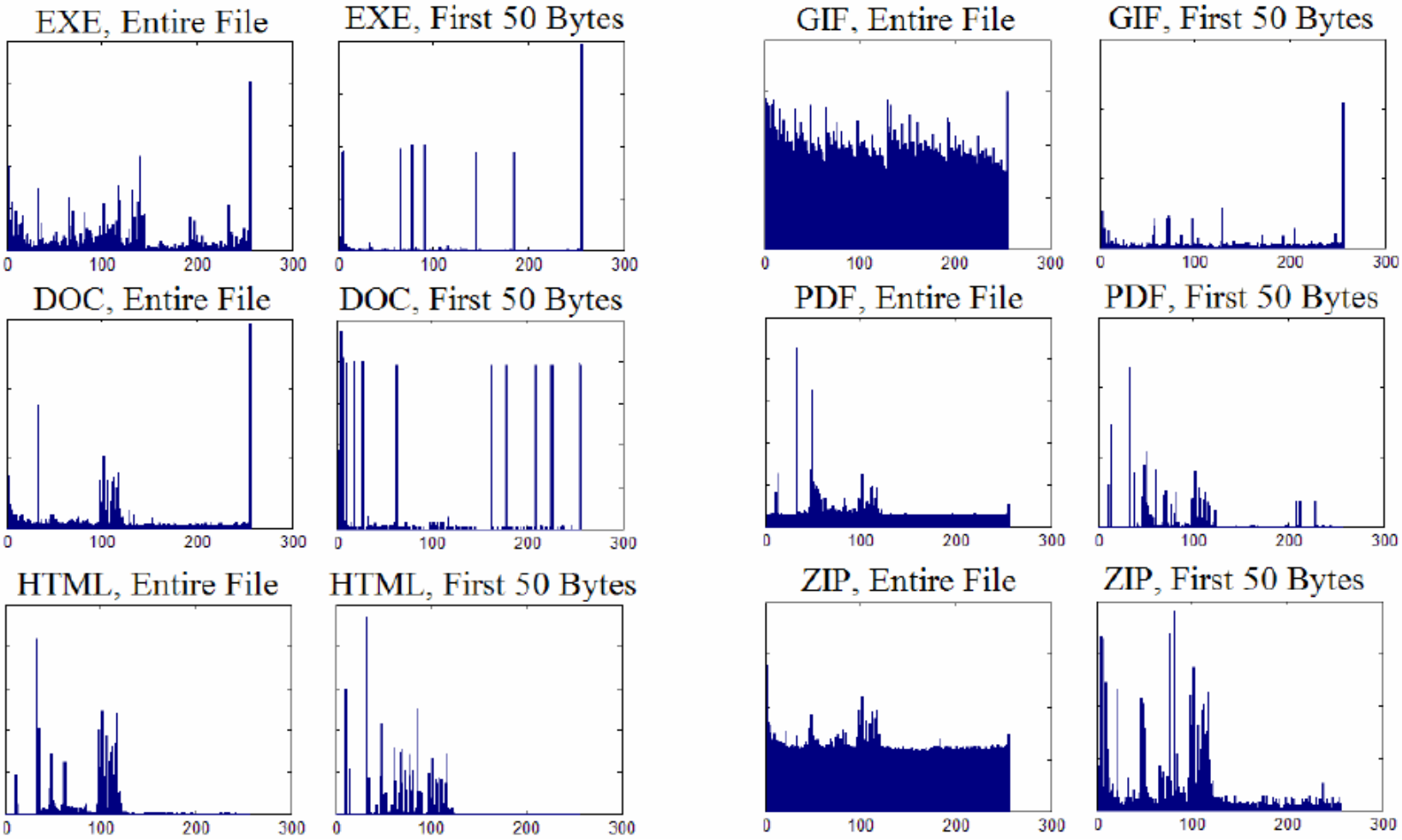
- N-GRAM 바이트 빈도 측정



Sliding window (window size = 3) ➔ 3-GRAM

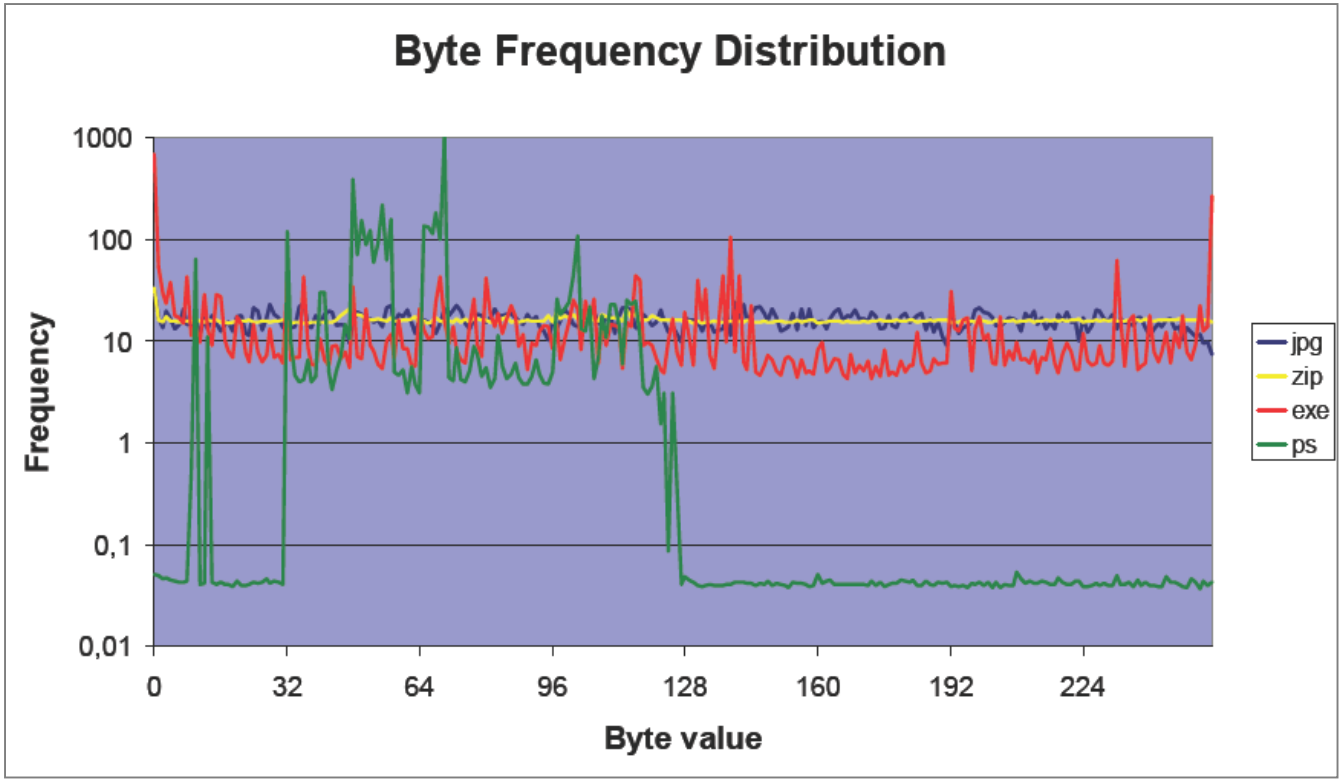
비연속적인 카빙

- 1-GRAM 바이트 빈도 측정



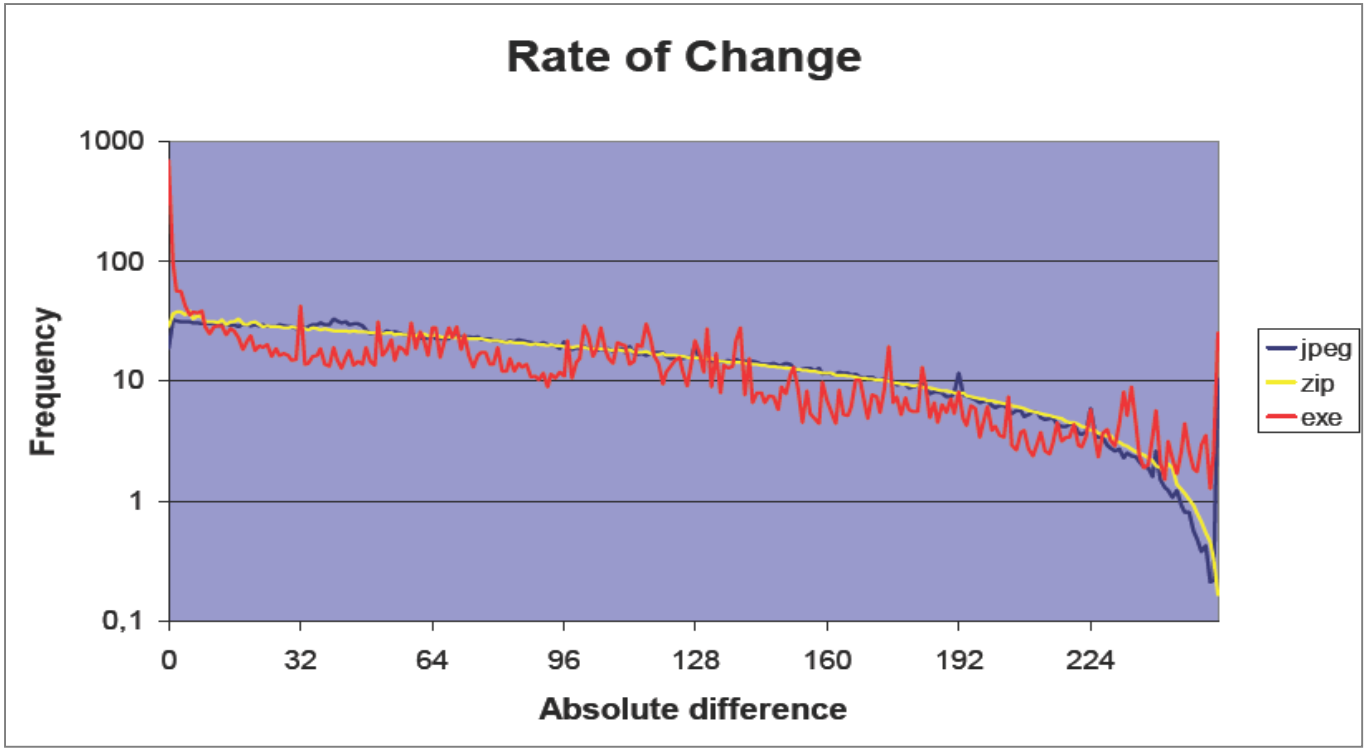
비연속적인 카빙

- **1-GRAM 바이트 빈도 측정**
  - 파일 별 평균인 기준점(Centroids) 선정 ➔ 기준점과의 차이를 측정 ➔ 허용치 판단



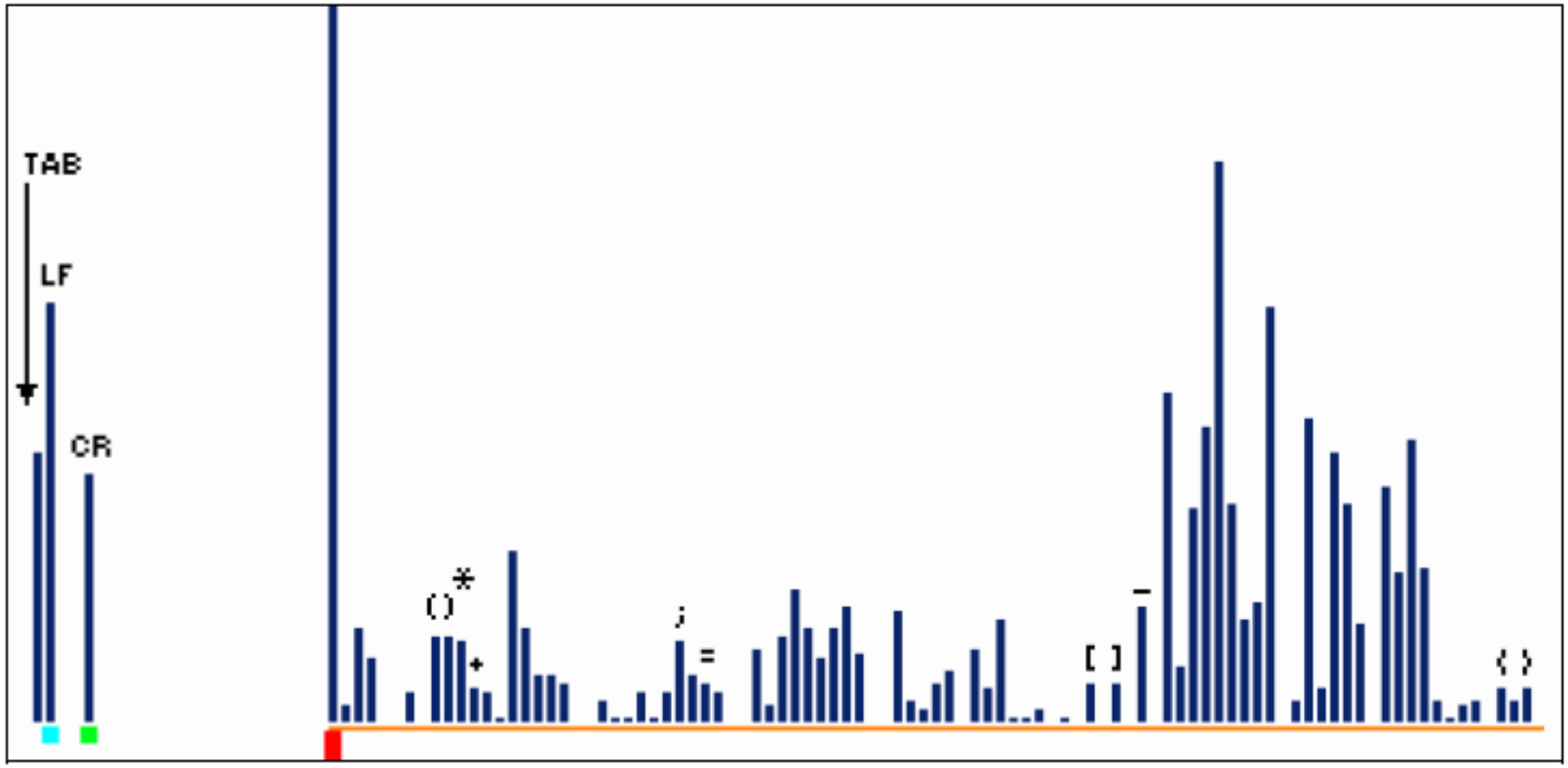
비연속적인 카빙

- **바이트 편차를 이용한 기법**
  - 인접한 바이트간의 편차를 이용한 방식



비연속적인 카빙

- **문자 빈도를 이용한 기법**
  - Ex) C++ 파일



삭제된 파일 복구

실습 #1

1. 시스템 드라이브에서 상주(Resident) 파일 하나 삭제 후 복구
2. 시스템 드라이브에서 비상주(Non-Resident) 파일 하나 삭제 후 복구
3. 시스템 드라이브에서 비상주(Non-Resident) 파일 여러 개 삭제 후 복구
4. 데이터 드라이브에서 비상주(Non-Resident) 파일 하나 삭제 후 복구
5. 외부저장장치에서 비상주(Non-Resident) 파일 하나 삭제 후 복구

삭제된 파일 복구

실습 #2

1. **코드게이트 2010 예선 – #1, #4** (http://forensic-proof.com/archives/418)
   - **#1** – MCDF 파일에서 카빙 (헤더/푸터를 이용한 시그니처 카빙)
   - **#2** – 비할당 영역에서 카빙 (헤더/푸터를 이용한 시그니처 카빙)

2. **코드게이트 2011 YUT Quals – Forensics 200** (http://forensic-proof.com/archives/1597)
   - **F200** – 비할당 영역에서 비연속적인 JPEG 카빙 (연속된 데이터, 엔트로피, 시그니처 기법)

3. **코드게이트 2011 YUT Challenge – 500**
   - **Y500** – 조각난 압축 파일 카빙 (엔트로피, 압축 블록 검증)

데이터 영구 삭제 기법

데이터 영구 삭제 기법

영구 삭제 (Wiping/Sanitization/Shredder/Secure Erase/Destruction) 방식

- **덮어쓰기 (Overwriting)**
  - 삭제하고자 하는 데이터에 [0, 1, Random]를 덮어씀
- **암호화 (Encryption)**
  - 안전한 방식을 이용해 디스크 혹은 파일 암호화
- **디가우징 (Degaussing)**
  - 강한 자기장에 노출시켜 자기디스크의 표면의 자력 흐름 파괴
- **물리적 파괴, 천공, 파쇄 (Physical Destruction)**
  - 강력한 기계를 사용해 물리적으로 구멍을 내거나 파쇄

데이터 영구 삭제 기법

디가우징 vs. 물리적 파괴

덮어쓰기 기반의 영구 삭제 표준

표준	년도	반복	패턴	비고
U.S. Navy	1993	3	문자, 보수, 랜덤	검증 필수
U.S. Air Force	1996	4	0, 1, 문자	검증 필수
Peter Gutmann	1996	1-35	매우 다양	원래 현재는 사용되지 않는 MFM, RLL을 위해
Bruce Schneier	1996	7	0, 1, 5번의 유사 랜덤	-
U.S. DoD	2001	3	문자, 보수, 다른 패턴	-
German Federal	2004	2-3	불규칙 패턴, 보수	-
CSEC	2006	3	0(1), 보수	분류되지 않은 매체를 위해
NIST	2006	1	?	-
U.S. NISP	2006	?	?	더 이상 지정하지 않음
NSA/CSS	2007	0	?	디가우즈 또는 파괴
Australian	2008	1	?	디가우즈 또는 일급 비밀 매체 파괴
New Zealand	2008	1	?	기밀 데이터를 위해

http://en.wikipedia.org/wiki/Data_erasure

데이터 영구 삭제 기법

덮어쓰기 방식

- **디스크 영역 영구 삭제**
  - 물리 섹터 시작~마지막 까지 덮어쓰기
  - 비할당 영역, 슬랙 공간, HPA, DCO 고려
- **파일 단위 영구 삭제**
  - 파일 데이터를 안전하게 영구 삭제
  - 메타데이터와 관련 아티팩트 (프리패치, 레지스트리 등)도 고려
- **표준 영구 삭제 기법을 지원하는 자동화 도구와 흔적 삭제 도구 사용**

데이터 영구 삭제 기법

덮어쓰기 기반의 영구 삭제 도구

- **BCWipe** (file, folder, free space, windows artifacts, file slack)
- **Hardwipe** (file, drive, space)
- **Eraser** (file, folder, free space, slack space)
- **CCleaner** (windows artifacts)
- **File Shredder** (file, folder)
- **SDelete** (file, folder, free space)
- **Darik's Boot And Nuke** (drive)
- **dd** (*nix) (file, drive)
- … …

실습 #3

1. Eraser로 파일 삭제 후 변화

2. Moo0 File Shredder로 파일 삭제 후 변화

영구 삭제 후 복구 가능성

영구 삭제 후 복구 가능성

피터 구트만 논문 (Peter Gutmann's Paper)

- **Secure Deletion of Data from Magnetic and Solid-State Memory (1996)**
  - MFM(Magnetic Force Microscopy)로 복구 가능성 언급
  - 0에다 1을 덮어쓰면 0.95, 1에다 1을 덮어쓰면 1.05 형태의 값이 기록 ➔ 이전 데이터 유추 가능
  - 기록할 때 쓰기 헤더 위치의 오차 발생 ➔ 트랙 가장자리의 잔여 데이터 남김
  - 최대 35번의 덮어쓰기가 필요 ➔ MFM, (1,7) RLL, (2,7) RLL 모두 대상
  - PRML 방식을 사용하는 디스크는 랜덤 데이터를 몇 번만 덮어써도 충분

영구 삭제 후 복구 가능성

피터 구트만 덮어쓰기 패턴 (35번)

- **고려된 인코딩**
  - MFM
  - (1,7) RLL
  - (2,7) RLL

Overwrite Data				
Pass No.	Data Written	Encoding Scheme Targeted		
1	Random			
2	Random			
3	Random			
4	Random			
5	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
6	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
7	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
8	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
9	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
10	00000000 00000000 00000000 0x00	(1,7) RLL	(2,7) RLL	
11	00010001 00010001 00010001 0x11	(1,7) RLL		
12	00100010 00100010 00100010 0x22	(1,7) RLL		
13	00110011 00110011 00110011 0x33	(1,7) RLL	(2,7) RLL	
14	01000100 01000100 01000100 0x44	(1,7) RLL		
15	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
16	01100110 01100110 01100110 0x66	(1,7) RLL	(2,7) RLL	
17	01110111 01110111 01110111 0x77	(1,7) RLL		
18	10001000 10001000 10001000 0x88	(1,7) RLL		
19	10011001 10011001 10011001 0x99	(1,7) RLL	(2,7) RLL	
20	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
21	10111011 10111011 10111011 0xBB	(1,7) RLL		
22	11001100 11001100 11001100 0xCC	(1,7) RLL	(2,7) RLL	
23	11011101 11011101 11011101 0xDD	(1,7) RLL		
24	11101110 11101110 11101110 0xEE	(1,7) RLL		
25	11111111 11111111 11111111 0xFF	(1,7) RLL	(2,7) RLL	
26	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
27	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
28	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
29	01101101 10110110 11011011 0x6D 0xB6 0xDB		(2,7) RLL	
30	10110110 11011011 01101101 0xB6 0xDB 0x6D		(2,7) RLL	
31	11011011 01101101 10110110 0xDB 0x6D 0xB6		(2,7) RLL	
32	Random			
33	Random			
34	Random			
35	Random			

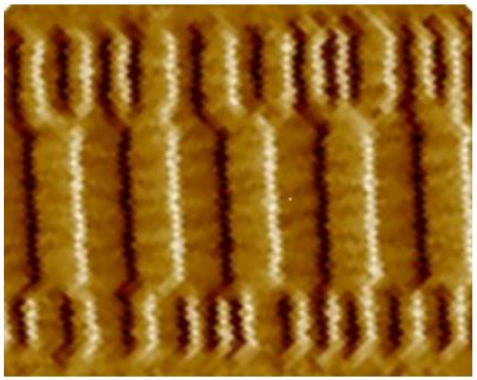
영구 삭제 후 복구 가능성

또 다른 논문

- **Overwriting Hard Drive Data: The Great Wiping Controversy (2008)**
  - 복구가 불가능하게 덮어쓰려면 몇 번 시도를 해야 하는지에 대한 논란이 많음
  - 피터 구트만 논문 이후 데이터를 한번만 덮어쓰면 복구할 수 있다는 논란이 커짐
  - 다수의 덮어쓰기는 너무 많은 시간이 요구됨
  - 이런 논란을 검증하고자 테스트 해봄 ➔ 포렌식적으로 의미가 있는지?
  - **잘못된 오해**
    - ✓ 1을 기록할 때, 0을 덮어쓰면 "0.95", 1을 덮어쓰면 "1.05"에 근접한다는 추정이 맞는가?
    - ✓ 각 쓰기 작업 시 정확한 "1"의 값을 쓰는 것이 가능한가?

영구 삭제 후 복구 가능성

MFM (Magnetic Force Microscopy)



- **MFM 기능**
  - 자기력에 의한 샘플 표면의 공간적 변이를 형상화
  - **1 바이트**를 복구하는데 **4분** 소요
  - 형상화로 알 수 있는 정보
    - ✓ 트랙의 너비와 왜곡
    - ✓ 전이 이상
    - ✓ 디스크 상의 기록 영역과 덮어쓴 영역의 차이

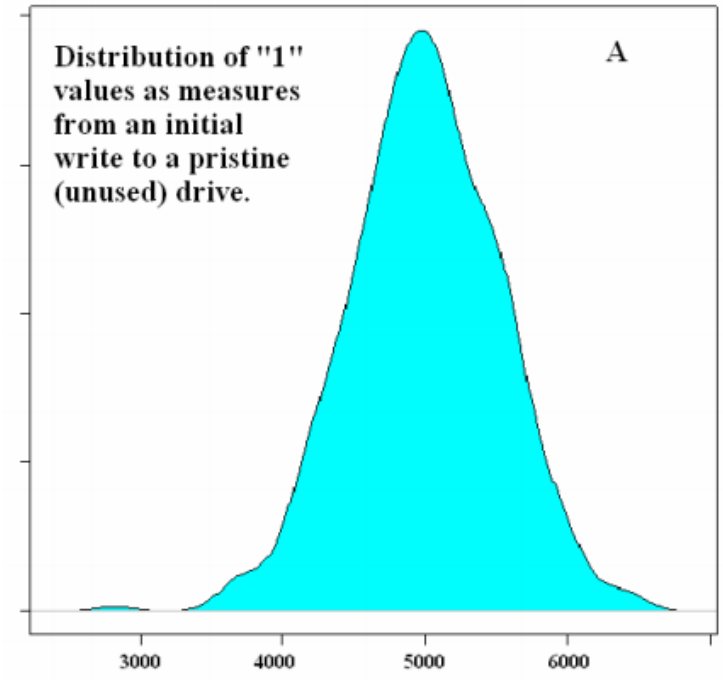
영구 삭제 후 복구 가능성

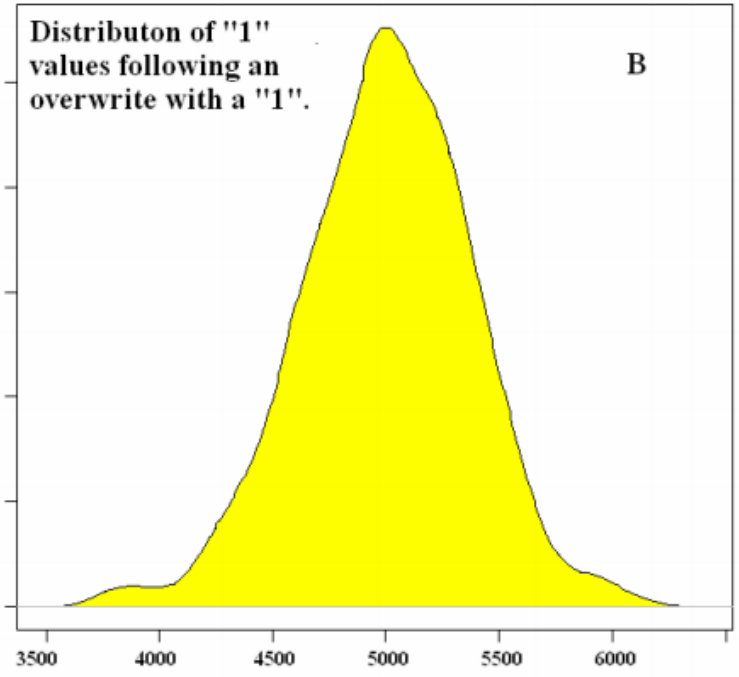
MFM (Magnetic Force Microscopy)

- **형상화 패턴의 변이**
  - **기록 패턴에 영향을 미치는 요인**
    - ✓ 헤드의 움직임 ➔ 내부적으로 진동, 공기 흐름 발생 ➔ 항상 정확한 곳에 위치?
    - ✓ 온도 ➔ 디스크 회전에 따른 내부 열로 자기력 변동 ➔ 플래터 온도가 올라가면 자기력이 감소됨
    - ✓ 랜덤 에러 ➔ 수많은 컴포넌트의 동작 중 발생하는 오류
    - ✓ 기록된 이전 데이터
  - **보완책**
    - ✓ 열 보정(Thermal Recalibration) 알고리즘을 사용하여 디스크 수축/팽창에 따른 속도 차이 개선
    - ✓ 알고리즘 개선, 내부 부품 개선, 인코딩 스키마 개선
  - 다양한 노력에도 이력 현상과 많은 외부 요인에 의해 동일한 패턴을 기록하기는 어려움

밀도 분포

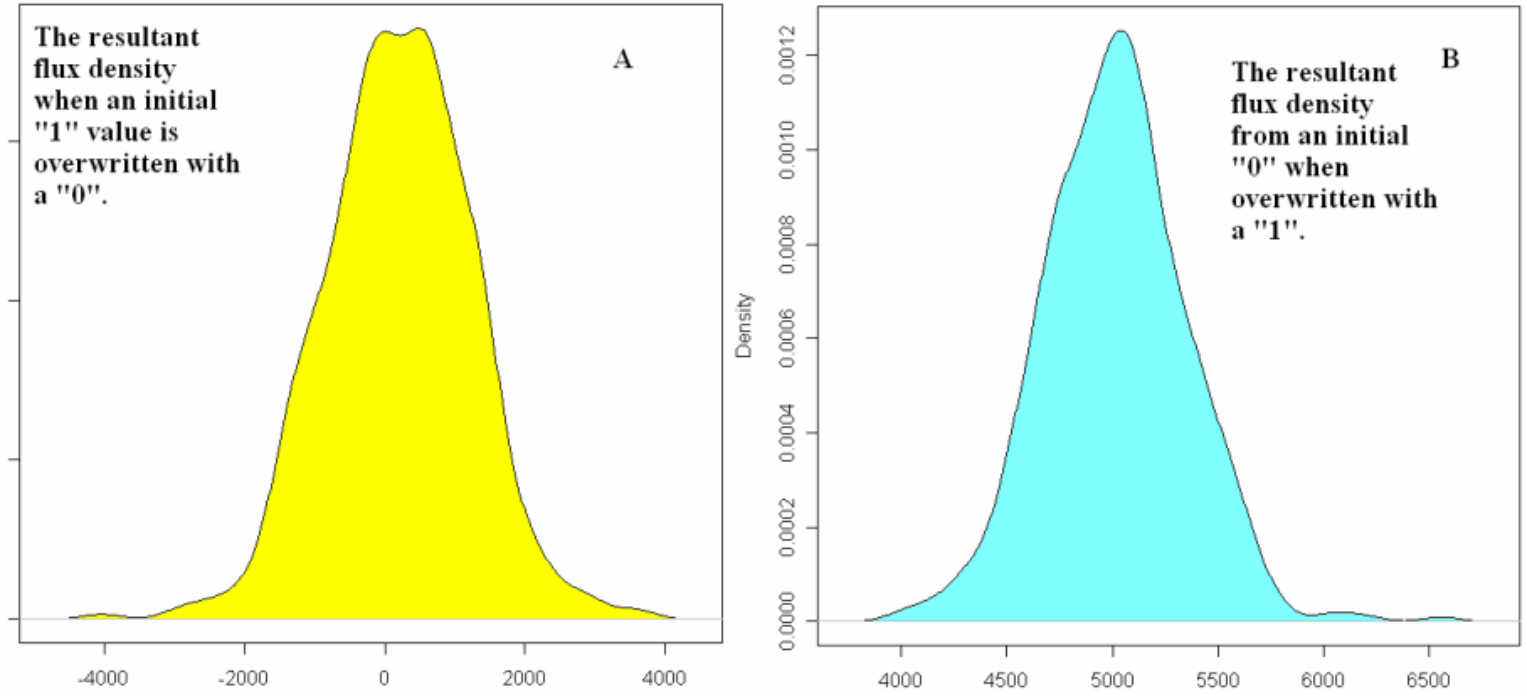
- **밀도 분포의 차이**
  - 디스크에 "1"을 기록하는 매 기록 작업마다 정확히 "1" 값이 기록될 수 있는가?
  - 온도 변동, 습도, 진동, 시간에 따른 부식으로 인해 덮어쓸 때마다 밀도 분포가 다름
  - 미세한 차이지만 값을 예측하기는 어려움





밀도 분포

- **밀도 분포의 차이**
  - "1.06"이라는 값을 얻었을 때, 이전 값의 영향인가? 온도의 영향인가? ➔ 판단 어려움
  - 시간이 지남에 따라 드라이브 자기장은 부식 (약해짐?)



영구 삭제 후 복구 가능성

복구 가능성

- **깨끗한 디스크 또는 영구 삭제된 디스크에서의 복구 가능성은 낮음**
  - 0.01% 보다 더 작을 수 있음
- **사용하고 있는 디스크(포맷된 디스크 포함)에서의 복구 가능성도 낮음**
  - 가능성은 좀 더 높지만 의미 있는 단어를 얻어낼 가능성은 적음
- **전자 현미경 기술이 더 발전한다면?**
  - 전자 현미경의 에러는 하드디스크 에러보다 더 적음
  - 전자 현미경의 기술적 한계가 아닌 하드디스크의 기술적 한계

영구 삭제 후 복구 가능성

에러 관리 로직

- **디스크 컨트롤러의 에러 최소화 기법**
  - **ECC Error Detection :** 섹터 Servo 영역에 저장된 ECC 활용
  - **ECC Error Correction :** ECC 에러 탐지 시 오류 정정 방식으로 복구
  - **Automatic Retry :** 갑작스런 움직임이나 온도 변화로 정확한 위치를 못 찾을 때, 정정 후 재시도
  - **Advanced Error Correction :** 고급 에러 정정 알고리즘 사용 ➔ 속도 느림
  - **Failure :** 섹터를 읽을 없는 경우, 에로 복구가 불가능

- 제조사마다 공통적인 에러 최소화 기법 사용 ➔ 에러의 영향은 줄어듦
- 현재 인코딩 스키마(PRML, EPRML)에서는 아날로그 값의 허용이 넓음 ➔ 굳이 보안책을
- 결과적으로, 이전 값을 알아내는 것은 확률 게임

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **카테고리 A**
  - 사용되지 않은 깨끗한 디스크
  - 포맷된 디스크 (NTFS의 기본 섹터 크기를 이용해 한번 포맷)
  - 시뮬레이션 드라이브 (랜덤한 데이터를 32번 덮어쓰기, /dev/random) ➔ 0으로 덮어씀

- **카테고리 B**
  - 최초 기록과 연속된 덮어쓰기에 5가지 패턴 사용
    - ✓ 모두 0
    - ✓ 모두 1
    - ✓ 01010101 패턴
    - ✓ 00111011 패턴
    - ✓ 00001111 패턴

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **17개의 선정된 모델 사용**
  - 오래된 Quantum 1GB에서 2006년 출시된 모델까지 (SCSI, IDE 등)
  - 56개의 하드디스크 테스트

- **실험**
  1. 1KB 파일을 이용해 데이터 기록
  2. 드라이브 왜곡과 비트는 모두 읽음
  3. 76,800 데이터 포인트 분석을 위해 각 절차를 5번 반복함
     - ✓ 사전 분포를 이용해 베이즈 정리(Bayes' Theorem)를 사용
     - ✓ 실제 포렌식 업무에서는 사전 데이터를 알 수 없음

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **오래된 드라이브 모델에 대한 확률 분포 테이블**
  - 초기 1을 기록한 후, 0으로 덮어씀 (이상적인 상황)

Probability of recovery	Pristine drive	Used Drive (ideal)
1 bit	0.92	0.56
2 bit	0.8464	0.3136
4 bit	0.71639296	0.098345
8 bits[5]	0.51321887	0.009672
16 bits	0.26339361	9.35E-05
32 bits	**0.06937619**	**8.75E-09**
64 bits	0.00481306	7.66E-17
128 bits	2.3166E-05	5.86E-33
256 bits	5.3664E-10	3.44E-65
512 bits	2.8798E-19	1.2E-129
1024 bits	8.2934E-38	1.4E-258

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **새로운 드라이브 모델에 대한 확률 분포 테이블**
  - 초기 1을 기록한 후, 0으로 덮어씀 (이상적인 상황)
  - 추가 덮어쓰기를 1회, 3회한 후 결과 비교

Probability of re-covery	Pristine drive (plus 1 wipe)	Pristine drive (plus 3 wipe)
1 bit	0.87	0.64
2 bit	0.7569	0.4096
4 bit	0.57289761	0.16777216
8 bits	0.328211672	0.028147498
16 bits	0.107722901	0.000792282
32 bits	**0.011604223**	**6.2771E-07**
64 bits	0.000134658	3.9402E-13
128 bits	1.81328E-08	1.55252E-25
256 bits	3.28798E-16	2.41031E-50
512 bits	1.08108E-31	5.8096E-100
1024 bits	1.16873E-62	3.3752E-199

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **새로운 드라이브 모델에 대한 확률 분포 테이블**
  - 고밀도의 EPRML 사용 드라이브의 복구 확률은 **어림 짐작 확률과 유사**
  - **2006년 모델 테스트**
    - ✓ 모두 0으로 와이핑된 디스크에 1을 덮어썼을 때 ➔ 최대 49.18%(+/- 0.11) 복구 확률
    - ✓ 모두 0으로 와이핑된 디스크에 다른 패턴 ➔ 최대 36.08%(+/- 0.24) 복구 확률
    - ✓ 일반적으로 사용하던 디스크의 복구 확률은?

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **복구 데이터 분포**

  - 8 비트를 정확히 읽었을 때, "1"로 표시

```
 [1]   0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1
[48]   0 1 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 0 0
[95]   0 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 0 1 0 0 1 1
[142] 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1 0 0
[189] 1 0 1 1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 0 1 0 1
[236] 0 0 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1
[283] 0 1 0 1 0 1 1 0 0 0 1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 1 1 0 0 0 0
[330] 0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 1 0 0 1 0 1 0
[377] 1 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1
[424] 1 1 0 0 1 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 1
[471] 1 1 1 1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0
[518] 1 0 0 1 0 1 1 1 1 1 0 0 1 0 1 1 0 1 0 1 1 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0
[565] 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0
[612] 1 1 0 1 1 1 1 0 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0
[659] 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 1 0 0 1 1
[706] 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 1
[753] 1 0 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 0
[800] 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0
[847] 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1 0 0 1 0
[894] 1 1 1 0 1 1 0 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1
[941] 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 0
[988] 0 0 1 0 1 1 1 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0
```

복구 가능성 테스트

- **예제 테스트**

  - **테스트 데이터**

> *Secure deletion of data - Peter Gutmann - 1996*
> *Abstract*
> *With the use of increasingly sophisticated encryption systems, an attacker wishing to gain access to sensitive data is forced to look elsewhere for information. One avenue of attack is the recovery of supposedly erased data from magnetic media or random-access memory.*

  - **복구 결과 (최선의 방법)**

> *‰*
> *‘cKræ}d8Œeti²n•of0daÊI0Ptr0G§tWÇíï_¼Á1u960eb8tÈñutW00000Dç•Ã#Ì0*
> *Hf$00¦000%£z0\0ã0000á0áä«it|tþÛ0u³e•Fƒºi™%|eàsinqTyøîopÚ”Ë:i†aze0*
> *®Mcryption0sîÙtems?DKtA""cÐÏ0+¢sinŒ0toK–ai2z ÷c(ns~0tü0;e*
> *½iti)e""daÆa>s0foôce¸ÑtÒÍl2o–*
> *ìeII¶˜$eöe›Ÿr""inf¬rm‰ion.0OnRïavem>egoN0- ¨tRÁ"1i*
> *läßh ±0"eÛoie=y0Czsu• ¨s/`lÜ{era’Jd0dataF ¨ro>•magne³;&£õãÈáã%or*r‰ndoª-Qcc«ÇŸ0mà*
> *@ryl000000000000000000*

영구 삭제 후 복구 가능성

복구 가능성 테스트

- **결과적으로...**
  - 개별 비트의 복구 가능성은 있지만 **의미 있는 데이터는 복구 불가능**
  - 디스크 제조사나 디스크 상태에 따라 차이 발생 ➔ **표준화하기는 어려움**
  - GB/TB의 정보를 자동으로 검색하는 **도구를 개발하는 것은 불가능**
  - 오래된 드라이브에서 조차도 **가능성이 낮음**
  - 법정에서 **증거로 활용하기에는 부적합**
  - 덮어쓴 드라이브에서의 복구 가능성에 대한 **논란은 종식시킬 필요가 있음**

질문 및 답변