GPS 조작 리포트

BoB 6기 디지털 포렌식 트랙

정경주

2017.07.18.

1. 사진

Figure 1 BoB 6기 디지털 포렌식 6기 정경주

2. JPG 파일에 들어가 있는 GPS 관련 정보들이 무엇이 있고, 어떤 형식으로 저장되어 있는지. 자신의 사진에서 실제 GPS가 있는 파일에서의 Offset을 구하기

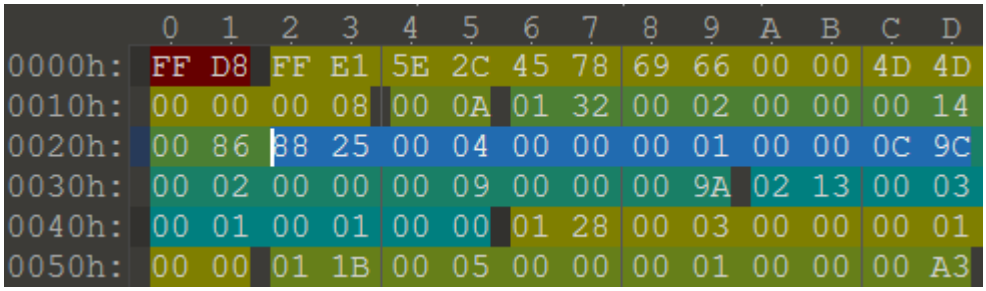
파일을 분석하기 위하여 010 Editor라는 툴을 사용하였다. JPG는 기본적으로 JPEG의 줄임말로 마커의 단위로 이루어져 있다. 이 구조에는 대표적으로 SOI(Start of Image), APP1(Application1), EOI(End of Image) 등으로 이루어져 있다. 이번 보고서에서 봐야할 부분은 APP1에 들어있는 exif의 포멧과 tiff의 포멧이다.

SOI는 보통 FF D8로 이루어져 있고, EOI는 FF D9로 이루어져 있다. 그 다음부터는 APP1이 나오는데 이 안에 Exif와 Tiff의 헤더가 들어가 있다. 45 78 69 66 00 00 4D 4D 가 이 두개의 헤더를 나타내는 것인데 45 78 69 66 00 00이 Exif의 헤더이고 4D 4D가 흔히 쓰이는 TiFF 이다.

4D 4D가 위치해 있는 곳은 C인데 데이터의 값을 구할때는 Offset + C의 값을 구한 위치에 가서 확인을 하면 된다.

APP1안에 있는 Image File Directory에는 GSPInfo(Tag = 0x8825가 들어있다). GPS에는 보통 우리가 생각할 때 경도, 위도, 고도로 위치를 파악 할 수 있다.

안에는 각각 12바이트의 엔트리가 있다. GPSInfo안을 들여다 보면 아래와 같다.]



	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0000h:	FF	D8	FF	E1	5E	2C	45	78	69	66	00	00	4D	4D
0010h:	00	00	00	08	00	0A	01	32	00	02	00	00	00	14
0020h:	00	86	88	25	00	04	00	00	00	01	00	00	0C	9C
0030h:	00	02	00	00	00	09	00	00	00	9A	02	13	00	03
0040h:	00	01	00	01	00	00	01	28	00	03	00	00	00	01
0050h:	00	00	01	1B	00	05	00	00	00	01	00	00	00	A3

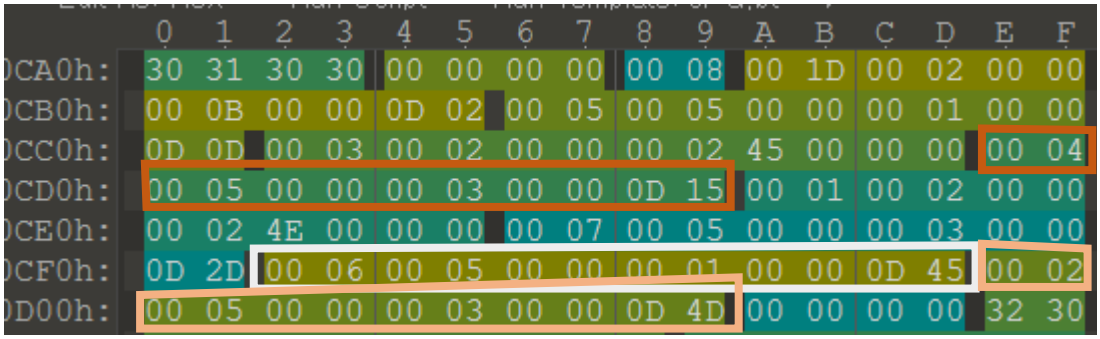
GPSInfo(0x8825) – 태크 번호(0x8825 2바이트) - 변수 타입(0x0004 2바이트) - 구성요소의 수(0x00000001 4바이트) - OFFset(From TIFF 0x00000C9C 4바이트)

C+C9C = CA8 를 하여 GPSInfo로의 위치를 따라가면 아래와 같은 태그들이 나온다. 하지만 GPS 조작에 필요한 정보는 위도, 경도, 고도기 때문에 표를 조금 잘랐다.

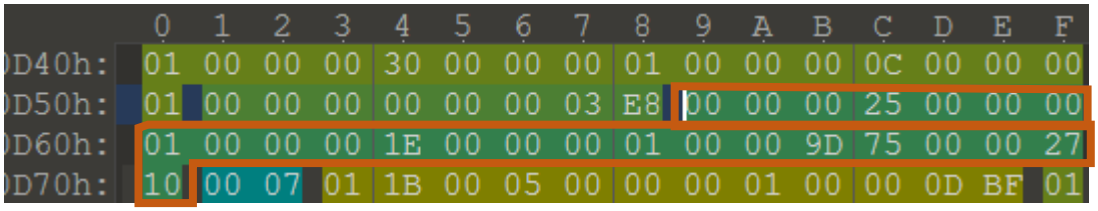
태그 ID	변수명	변수타입	크기	설명
0	GPSVersionID	BYTE	4	GPS version
1	GPSLatitudeRef	ASCII	2	N(북쪽), S(남쪽)
2	GPSLatitude	RATIONAL	3	위도
3	GPSLongitudeRef	ASCII	2	E(동쪽), W(서쪽)
4	GPSLongitude	RATIONAL	3	경도
5	GPSAltitudeRef	BYTE	1	고도와 해수면과의 관계
6	GPSAltitude	RATIONAL	1	고도
7	GPSTimeStamp	RATIONAL	3	UTC(세계협정시)로 변환한 GPS촬영시간(시,분,초)
8	GPSSatellites	ASCII	무관	측정에 사용된 위성
9	GPSStatus	ASCII	2	촬영에 사용된 GPS 수신기의 상태
10	GPSMeasureMode	ASCII	2	GPS 측정 모드
11	GPSDOP	RATIONAL	1	GPS 데이터의 정확도
12	GPSSpeedRef	ASCII	2	GPS 수신기 속도에 관한 부호

GPS INFO에서 필요한 태그인 2(GPSLatitude), 4(GPSLongitude), 6(GPSAltitude)의 값들을 보면 아래와 같다. 실제 데이터 위치를 표기 한 후에는 수정후의 위치도 표현을 하였다.

이 모든 값들은 Rational이라는 타입을 선언이 되는데 위도, 경도를 나타내기 위하여 총 8바이트 3개 즉 24바이트로 이루어져 있으며 각 8바이트에서 앞 4바이트 나누기 뒤 4바이트를 한 값을 '도'라고 한다. 그 다음 8바이트를 방금 한 것과 같이 계산하면 '분'이라고 하며 마지막 8바이트를 계산하면 '초'라고 한다. 고도도 같은 형식으로 구하지만 8바이트가 3개인 24바이트가 아니라 총 8바이트 한 개 인 총 8바이트로 이루어져 있다.



- **GPSLatitude 00 02(02) 00 05(RATIONAL) 00 00 00 03(크기) 00 00 0D 4D(N(북쪽), S(남쪽))**

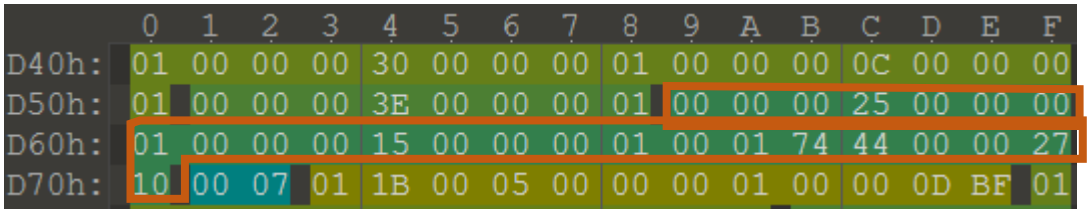


실제 데이터 위치: C + D4D = D59

00 00 00 25 / 00 00 00 01 / 00 00 00 1E / 00 00 00 01 / 00 00 9D 75 / 00 00 27 10

37/1; 30/1; 40309 / 10000 -> 37; 30; 4.0309;

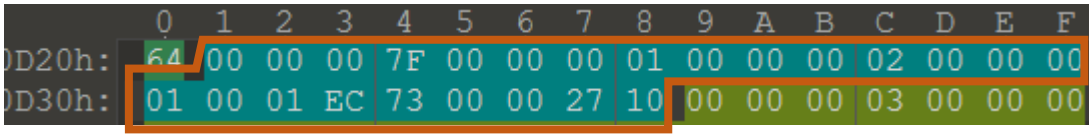
수정 후)



00 00 00 25 / 00 00 00 01 / 00 00 00 15 / 00 00 00 01 / 00 01 74 44 / 00 00 27 10

37/1; 21/1; 95300/10000 -> 37; 21; 9.53

- **GPSLongitude 00 04(4) 00 05(RATIONAL) 00 00 00 03(크기) 00 00 0D 15(E(동쪽), W(서쪽))**

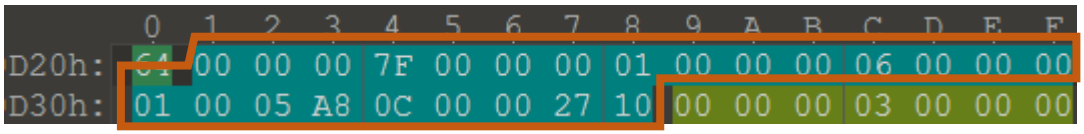


실제 데이터 위치: C + D15 = D21

00 00 00 7F / 00 00 00 01 / 00 00 00 02 / 00 00 00 01 / 00 01 EC 73 / 00 00 27 10

127/1; 2/1; 126,067/ 10000 -> 127; 2; 12.6067

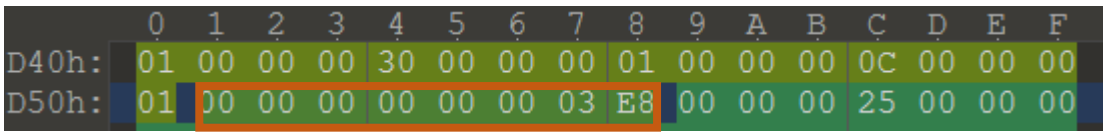
수정 후)



00 00 00 7F / 00 00 00 01 / 00 00 00 06 / 00 00 00 01 / 00 05 A8 0C / 00 00 27 10

127/1; 6/1; 370700/ 10000 -> 127; 6; 37.07

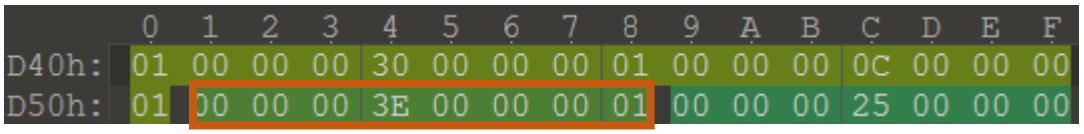
- **GPSAltitude 00 06(06) 00 05(RATIONAL) 00 00 00 01(크기) 00 00 0D 45(고도)**



실제 데이터 위치: C + D45 = D51

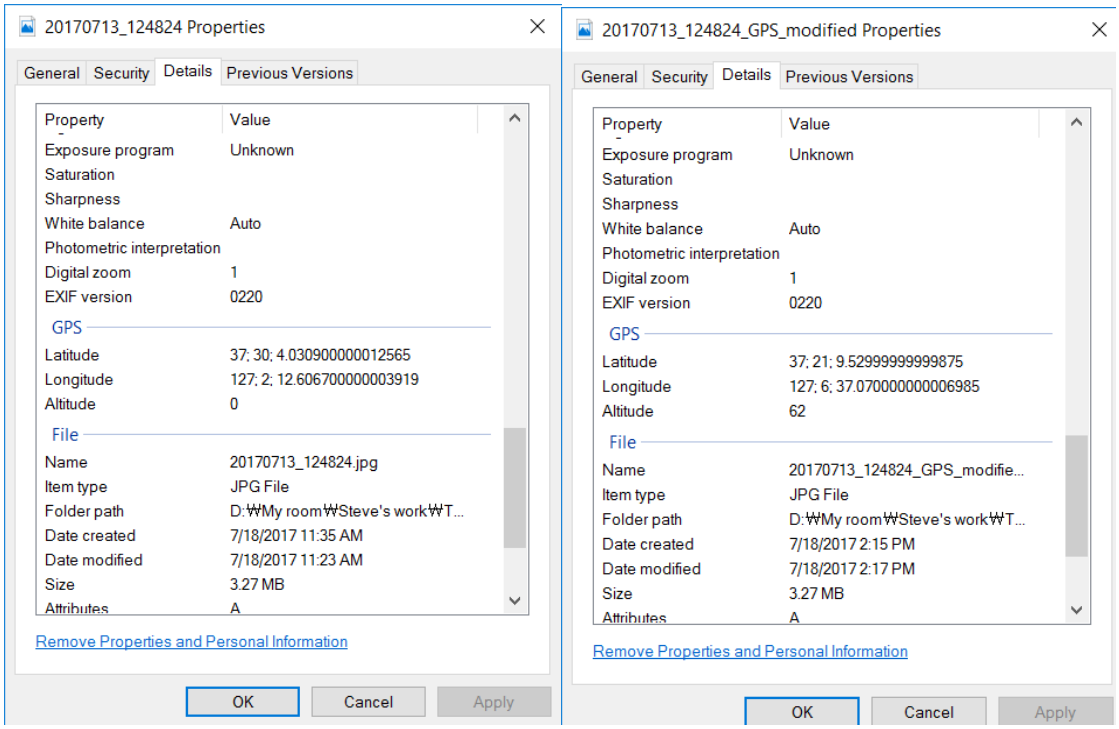
00 00 00 00 00 00 03 E8 -> 1000

수정 후)



00 00 00 00 00 00 00 3E->62

아래는 비교 사진입니다.



강남 비오비 센터에서 경기도 성남시 분당구 성원아파트로 옮겨졌습니다.

3. 이렇게 조작했을 경우에 이를 디지털 포렌식 관점에서 어떻게 찾을 수 있을것인지? 자신의 아이디어와 그에 대한 논리적인 근거를 적어주시면 됩니다.

먼저 조작 자체에 대한 수정을 알기 위해서는 수정 시간과 접근 시간을 비교해 보면 될 것 같다. 사진의 수정 시간과 접근 시간이 같은지를 보고 그렇다고 한다면 바이너리 디핑이라는 기술로 가지고 있는 두 사진의 값들을 비교를 해보는 것이다. 만약에 GPS정보가 다르다고 한다면 이는 GPS 정보가 조작 됐다는 것으로, 법정에서 아무 효력을 발휘할 수 없을 것이다.