

G-File에 저장된 위치정보 관리 알고리즘

G-File stored in the location information management algorithm

최상균*

Sang-Kyoon Choi*

요 약

G-File은 사진촬영의 위치 좌표를 내장한 사진 파일을 이용하여 사용자에게 사진 속 이미지의 위치와 방향각 데이터를 이용한 파일을 말한다[1]. G-File은 사진 파일을 입력받아 저장하는 사진 파일 수신 수단과, 사진 파일을 사진 정보와 위치 정보로 분리하여 피사체의 위치 정보를 추출하는 위치 정보 분석 수단 및 위치 정보를 이용하여 지도상의 대응되는 좌표에 표시하는 위치 안내 수단을 포함하는 것을 특징으로 한다. 본 논문은, 이러한 G-File에 저장된 위치 정보에 대한 관리를 할 수 있는 알고리즘을 제안한다. 이 알고리즘은 G-File을 이용하는 사용자에게 정보를 관리하게 하고, G-File 관리에 편리함을 제공하기 위함이다.

Abstract

G-File on the position of the photo shoot pictures with built-in picture file to the user, the location and orientation of the image files using the data refers to the [1]. G-File is to store photo files, photo files and receiving means for receiving input, and picture files, photos and location information extracted by separating the subject's location information with location information and location analysis by means of the corresponding coordinates on the map including the location indicated in the guide means is characterized in that. In this paper, the G-File on the location information stored in the algorithm that can be managed is proposed. G-File on the algorithm used to manage the information to the user, G-File management is to provide convenience.

Key words : G-File, Exif, GPS

I. 서 론

최근 지리공간에 대한 관심이 부각되고 있으며, 각종 포털사이트에서 지리공간 정보 서비스를 실시하고 있다[1],[2].

지리공간과 전자지도 서비스에 대한 관심이 부각되면서 매쉬업 개발이 많이 이루어지고 있다. 매쉬업을 통해 다양한 부가 정보가 지도와 결합되고 있다.

[1],[2].

본 연구에서는 JPEG방식과 Exif와 매쉬업에 관련된 내용을 이용하여 G-File 시스템을 이용하여 개발된 GPIR(GPS Position Information Revision)의 데이터를 변환 및 출력 등을 할 수 있는 관리 알고리즘을 제안한다. 지금까지는 G-File을 개발한 연구진의 관리 모듈에 의하여 관리되던 G-File 위치정보를 일반 이용자에게도 공개하여 이를 관리할 수 있도록 하기

* 김포대학 e-비즈니스과 부교수
· 제1저자 (First Author) : 최상균
· 투고일자 : 2011년 8월 23일
· 심사(수정)일자 : 2011년 8월 23일 (수정일자 : 2011년 9월 29일)
· 게재일자 : 2011년 10월 30일

위함이다.

본 논문의 구성은 다음과 같다. 2장은 본 논문에서 다루고 있는 기반기술인 Exif 기술에 대하여 논하였고, 3장은 G-File 시스템 및 GPIR 시스템에 대하여 논하였고, 4장은 G-File 관리 알고리즘을 논하였다. 마지막으로 5장 결론에서는 연구내용을 정리하고 향후 과제를 제안하였다.

II. 관련기술

2-1 Exif

Exif는 디지털 카메라에서 이미지와 관련한 파일 포맷에서 이미지에 대한 상세 정보를 추가하기 위해서 JEIDA(Japan Electronic Industry Development Association)에서 만들어졌으며 JPEG, TIFF Rev. 6.0, RIFF WAVE 파일 포맷이 지원되며, JPEG2000이나 PNG는 지원되지 않는다[1].

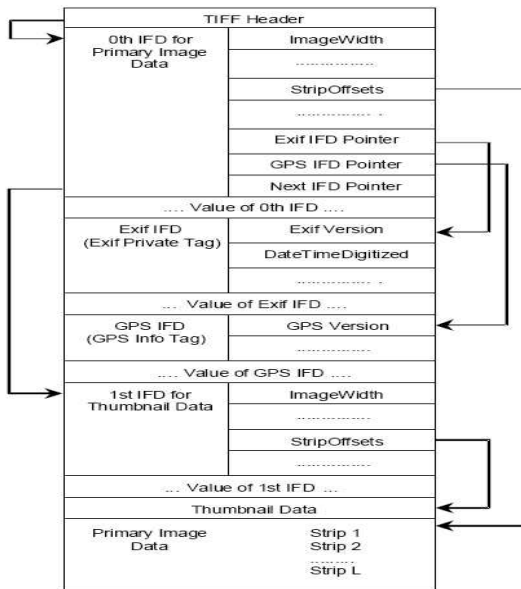


그림 1. 비 압축 파일 구조
Fig. 1. Non-compaction file architecture

Exif에 포함되는 정보로는 날짜와 시간 정보(Data and time information), 셔터 스피드, 발광모드 등과 같은 카메라 설정 정보(Camera settings information), 촬영된 지역정보(Location information), 요약 및 저작권

관련 정보(Descriptions and Copyright information)가 포함된다[3].

Exif의 파일구조는 그림 1과 같이 TIFF Header, Exif IFD(Image File Directories), GPS IFD, Image Data 등으로 구성된다[3],[4].

Exif 파일구조 가운데 헤더 부분인 TIFF Header에는 다음 IFD에 대한 정보를 포함하고 있으며, 각각의 IFD에 해당하는 데이터 포인터 정보를 가지고 있고, 전체 8Bytes로 구성되어 있다. TIFF Header 구조는 다음과 같다[3].

표 1. Tiff Headers
Table 1. Tiff Headers

Name	Size (Bytes)	Value
Byte Order5	2	Written as either "II" (4949.H) (little endian) or "MM" (4D4D.H) (big endian) depending on the CPU of the machine doing the recording.
42	2	002A.H (fixed)
Offset of IFD	4	0th IFD offset. If the TIFF header is followed immediately by the 0th IFD, it is written as 00000008.H.

III. GPIR 시스템

3-1 G-File

G-File은 사진촬영의 위치 좌표를 내장한 사진 파일을 이용하여 사용자에게 사진 속 이미지의 위치와 방향각 데이터를 이용한 파일을 말한다[1].

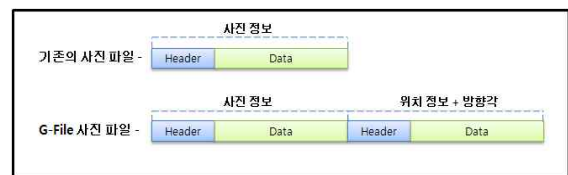


그림 2. G-File 시스템
Fig. 2. G-File system

G-File은 사진 속 사진촬영 위치 정보가 포함된 사진을 입력받아 지도상에서 위치를 표시하는 지리정보 안내 시스템에 있어서, 사진 파일을 입력받아 저장하는 사진 파일 수신 수단과 사진 파일을 사진 정

보와 위치 정보로 분리하여 피사체의 위치 정보를 추출하는 위치 정보 분석 수단 및 위치 정보를 이용하여 지도상의 대응되는 좌표에 표시하는 위치 안내 수단을 포함하는 것을 특징으로 한다[1].

[1]에서는 GPIR 시스템 설계의 상세 정보로 전자지도 매쉬업에 관한 내용으로 Exif 개발과 전자지도 매핑하는 방법 위치보정 방법 및 프로세스에 대하여 상세하게 설명하고 있다. 이를 이용하여 사진 촬영 시 사진촬영 위치를 파악하여 사진 파일에 함께 저장하고, 사용자가 사진을 보면서 그 위치가 궁금할 경우 사진 파일을 입력받아 바로 지도를 통하여 피사체의 위치를 확인할 수 있다.

Tag Name	Field Name	Tag ID		Type	Count
		Dec	Hex		
A. Tags Relating to Version					
Exif version	ExifVersion	36864	9000	UNDEFINED	4
Supported FlashPix version	FlashPixVersion	40960	A000	UNDEFINED	4
B. Tag Relating to Image Data Characteristics					
Color space information	ColorSpace	40961	A001	SHORT	1
C. Tags Relating to Image Configuration					
Meaning of each component	ComponentsConfiguration	37121	9101	UNDEFINED	4
Image compression mode	CompressedBitsPerPixel	37122	9102	RATIONAL	1
Valid image width	PixelXDimension	40962	A002	SHORT or LONG	1
Valid image height	PixelYDimension	40963	A003	SHORT or LONG	1
D. Tags Relating to User Information					
Manufacturer notes	MakerNote	37500	927C	UNDEFINED	Any
User comments	UserComment	37510	9286	UNDEFINED	Any
E. Tag Relating to Related File Information					
Related audio file	RelatedSoundFile	40964	A004	ASCII	13
F. Tags Relating to Date and Time					
Date and time of original data generation	DateTimeOriginal	36867	9003	ASCII	20
Date and time of digital data generation	DateTimeDigitized	36868	9004	ASCII	20
DateTimeSubseconds	SubSecTime	37520	9290	ASCII	Any
DateTimeOriginalSubseconds	SubSecTimeOriginal	37521	9291	ASCII	Any
DateTimeDigitizedSubseconds	SubSecTimeDigitized	37522	9292	ASCII	Any
G. Tags Relating to Picture-Taking Conditions					
See Table 5					
H. Tags Relating to Date and Time					
Pointer of Interoperability IFD	Interoperability IFD Pointer	40965	A005	LONG	1

그림 3. Exif 카메라 정보
Fig. 3. Camera information of Exif

G-File 시스템을 활용하여 G-File에 포함되어 있는 데이터를 나타낸 화면은 다음과 같다.

Image Description	270
Date and Time	306 2008:12:09 05:12:14
Artist	315
Manufacturer	271 Nikon
Model	272 COOLPIX P2
Software	305 Adobe Photoshop CS3 Windows
Orientation	274 top - left
Resolution Unit	296 Inch
x-Resolution	282 3000000/10000
y-Resolution	283 3000000/10000
Pixel x-Dimension	40962 300
Pixel y-Dimension	40963 225
Exif Version	36864 Exif Version 0.0
GPS Version Id	0 2/2/0/0
GPS Latitude Ref	1 N
GPS Latitude	2 121.3415, 5678100
GPS Longitude Ref	3 E
GPS Longitude	4 133.1, 45110, 6789100
GPS Dest Latitude Ref	19 N
GPS Dest Latitude	20 11.1, 22100, 3333100
GPS Dest Longitude Ref	21 E
GPS Dest Longitude	22 111.1, 11100, 2222100
GPS Angle	31 900
Contrast	
Sharpness	
Saturation	

그림 4. G-File 데이터 확인
Fig. 4. Data identification of G-File

3-2 GPIR 시스템

GPIR 시스템은 기존의 GPS 오차들을 해결 가능하고, 이용자가 가고자 하는 위치의 정보를 손쉽게 얻을 수 있다. 그리고 이러한 GPS의 오차에 대한 위치 보정을 이용자가 직접 수정하게 되며, 자동으로 데이터베이스에 저장됨으로써 보다 정확한 위치정보를 파악할 수 있게 된다[1].

이러한 GPIR 시스템의 원리는 피사체의 위치를 선택하게 되고, 진북과 피사체와의 각을 읽거나, GPS에서 위치정보를 제공받으면 위치를 보정 받을 수 있는 체계가 완성된다. GPIR 시스템은 사진촬영 위치 주변 10m 주변과 사진 촬영 방향으로 데이터베이스를 검색하여 리스트를 제공하여, 보다 정확한 피사체 위치를 제공하는 것을 목적으로 하는바, 위치보정은 이러한 정보에 대한 정확성을 높이는데 중요한 역할을 하게 된다[2].

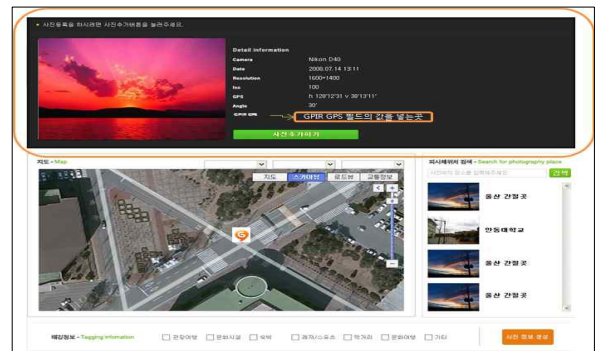


그림 5. GPIR 위치보정 작업
Fig. 5. Position revision of GPIR

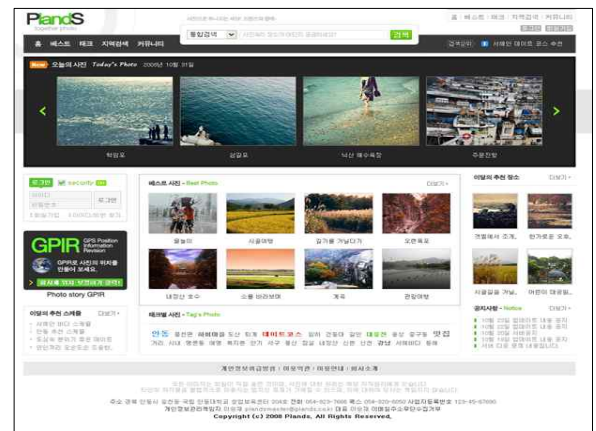


그림 6. GPIR 시스템 메인화면
Fig. 6. GPIR Main Menu

IV. G-File 위치정보 관리 알고리즘

G-File 수신기는 카메라 기능과 GPS 수신 기능이 복합된 휴대 단말에서 사진을 찍을 때 GPS 좌표를 수신하여 사진 데이터와 위치 데이터를 링크시켜 관리하는 기술을 대처하기 위함이다[2].

지금까지 G-File을 이용하는 GPIR 시스템에 저장된 데이터는 약 150,000건에 이르고 있고, 이용자의 수도 1,000명에 이르고 있다. 계속하여 GPIR 시스템을 이용하는 이용자가 증가함에 따라 GPIR에 저장된 G-File의 정보 관리에 이용자의 관심이 높아지고 있다. 따라서 GPIR 이용자들에게 G-File을 직접 관리할 수 있는 알고리즘을 제안하여 보급함으로써 GPIR 시스템을 이용하고 G-File을 관리함에 편리함을 제공한다. 위치정보 관리 알고리즘은 C++로 작성되었다 [5],[6].

다음의 표는 G-File 관리 알고리즘 가운데, Exif 메타데이터 정보를 추가, 수정 및 데이터를 삭제하는 방법을 보여주는 알고리즘이다.

표 2. Exif 정보 관리 알고리즘

Table 2. Algorithm of Exif information management

```
// ***** -*- C++
-*/
// Add, modify and delete Exif metadata.

#include <exiv2/exiv2.hpp>

#include <iostream>
#include <iomanip>
#include <cassert>

int main(int argc, char* const argv[])
try {
    if (argc != 2) {
        std::cout << "Usage: " << argv[0] << "
file\n";
        return 1;
    }
    std::string file(argv[1]);

    Exiv2::ExifData exifData;

    // *****
    // Add to the Exif data

    exifData["Exif.Image.Model"] = "Test 1";
    exifData["Exif.Image.SamplesPerPixel"] =
uint16_t(162);
    exifData["Exif.Image.XResolution"] =
int32_t(-2);
    exifData["Exif.Image.YResolution"] =
```

```
Exiv2::Rational(-2, 3);
    std::cout << "Added a few tags the quick
way.\n";

    Exiv2::Value::AutoPtr v =
Exiv2::Value::create(Exiv2::asciiString);

    v->read("2009:12:31 23:59:59");

    Exiv2::ExifKey key("Exif.Photo.DateTimeOriginal");
    exifData.add(key, v.get());
    std::cout << "Added key \'" << key << "\",
value \'" << *v << "\n";

    Exiv2::URationalValue::AutoPtr rv(new
Exiv2::URationalValue);

    rv->read("1/2 1/3");

    rv->value_push_back(std::make_pair(2,3));
    rv->value_push_back(std::make_pair(3,4));

    Exiv2::ExifKey key("Exif.Image.PrimaryChromaticities");
    exifData.add(key, rv.get());
    std::cout << "Added key \'" << key << "\",
value \'" << *rv << "\n";

    // *****
    // Modify Exif data

    Exiv2::Exifdatum& tag =
exifData["Exif.Photo.DateTimeOriginal"];
    std::string date = tag.toString();
    date.replace(0, 4, "2000");
    tag.setValue(date);
    std::cout << "Modified key \'" << key
<< "\", new value \'" <<
tag.value() << "\n";

    Exiv2::ExifKey key("Exif.Image.PrimaryChromaticities");
    Exiv2::ExifData::iterator pos =
exifData.findKey(key);
    if (pos == exifData.end()) throw
Exiv2::Error(1, "Key not found");
    v = pos->getValue();

    Exiv2::URationalValue* prv = dynam-
ic_cast<Exiv2::URationalValue*>(v.release());
    if (prv == 0) throw Exiv2::Error(1,
"Downcast failed");
    rv = Exiv2::URationalValue::AutoPtr(prv);

    rv->value_[2] = std::make_pair(88,77);

    pos->setValue(rv.get());
    std::cout << "Modified key \'" << key
<< "\", new value \'" <<
pos->value() << "\n";

    // *****
    // Delete metadata from the Exif data
```

```

        key =
Exiv2::ExifKey("Exif.Image.PrimaryChromaticities")
;
        pos = exifData.findKey(key);
        if (pos == exifData.end()) throw
Exiv2::Error(1, "Key not found");
        exifData.erase(pos);
        std::cout << "Deleted key \" << key <<
        "\\n";

        Exiv2::Image::AutoPtr image =
Exiv2::ImageFactory::open(file);
        assert(image.get() != 0);

        image->setExifData(exifData);
        image->writeMetadata();

        return 0;
    }
    catch (Exiv2::AnyError& e) {
        std::cout << "Caught Exiv2 exception \" <<
e << "\\n";
        return -1;
    }
}
    
```

표 2. Exif 정보 관리 알고리즘은 Exif 메타데이터 정보에 대한 추가로, 이미지 모델과 X해상도, Y해상도 정보를 추가하여 관리하게 된다. 또한 정보의 수정은 Key에 의하여 수정되는데, 이미지 속성의 변화는 없다. 삭제의 경우 메타데이터가 삭제되는데, Key에 의하여 삭제된다.

다음은 G-File을 관리하는 관리자는 파일의 속성에 대하여 주석(Comment)을 관리할 경우 사용하는 알고리즘이다. 이는 많은 양의 G-File을 관리함에 있어 편리함을 제공할 목적으로 작성하였으며, 대용량의 G-File 관리에도 좋은 효과가 있음이 자체 분석에서 입증되었다.

표 3. Exif 주석달기 알고리즘
Table 3. Algorithm of Exif comment

```

#include <exiv2/exiv2.hpp>

#include <iostream>
#include <cassert>

int main(int argc, char* const argv[])
try {
    if (argc != 2) {
        std::cout << "Usage: " << argv[0] << "
file\n";
        return 1;
    }

    Exiv2::Image::AutoPtr image =
Exiv2::ImageFactory::open(argv[1]);
    
```

```

        assert (image.get() != 0);
        image->readMetadata();
        Exiv2::ExifData &exifData = im-
age->exifData();

        exifData["Exif.Photo.UserComment"]
        = "charset=\"Unicode\" An Unicode Exif
comment added with Exiv2";
        exifData["Exif.Photo.UserComment"]
        = "charset=\"Undefined\" An undefined
Exif comment added with Exiv2";
        exifData["Exif.Photo.UserComment"]
        = "Another undefined Exif comment
added with Exiv2";
        exifData["Exif.Photo.UserComment"]
        = "charset=Ascii An ASCII Exif com-
ment added with Exiv2";

        std::cout << "Writing user comment \"
        <
exifData["Exif.Photo.UserComment"]
        << "\" back to the image\n";

        image->writeMetadata();

        return 0;
    }
    catch (Exiv2::AnyError& e) {
        std::cout << "Caught Exiv2 exception \" <<
e << "\\n";
        return -1;
    }
}
    
```

다음은 G-File을 관리하는 관리자는 Exif 파일의 속성을 출력하는데 사용하는 알고리즘이다. G-File 사용자는 자신이 관리하고 있는 G-File을 관리 및 전송 목적으로 Exif 파일의 속성을 출력하게 되는데 이는 파일 속성에 잘못 입력된 정보가 있으면 이를 걸러내는 효과도 본 알고리즘에서 제공하고 있기 때문에 양질의 데이터를 관리하는데 큰 도움이 될 것으로 기대된다.

표 4. Exif 출력 알고리즘
Table 4. Algorithm of Exif print

```

// ***** -*- C++ -*-
// Program to print the Exif metadata of an im-
age

#include <exiv2/exiv2.hpp>

#include <iostream>
#include <iomanip>
#include <cassert>

int main(int argc, char* const argv[])
try {
    if (argc != 2) {
        std::cout << "Usage: " << argv[0] << "
    
```

```

file\n";
    return 1;
}

Exiv2::Image::AutoPtr image =
Exiv2::ImageFactory::open(argv[1]);
assert(image.get() != 0);
image->readMetadata();

Exiv2::ExifData &exifData = im-
age->exifData();
if (exifData.empty()) {
    std::string error(argv[1]);
    error += ": No Exif data found in the
file";
    throw Exiv2::Error(1, error);
}
Exiv2::ExifData::const_iterator end =
exifData.end();
for (Exiv2::ExifData::const_iterator i =
exifData.begin(); i != end; ++i) {
    const char* tn = i->typeName();
    std::cout << std::setw(44) << std::set-
fill(' ') << std::left
        << i->key() << " "
        << "0x" << std::setw(4) <<
std::setfill('0') << std::right
        << std::hex << i->tag() <<
" "
        << std::setw(9) << std::set-
fill(' ') << std::left
        << (tn ? tn : "Unknown") <<
" "
        << std::dec << std::setw(3)
        << std::setfill(' ') <<
std::right
        << i->count() << " "
        << std::dec << i->value()
        << "\n";
}
return 0;
}

catch (Exiv2::Error& e) {
    std::cout << "Caught Exiv2 exception '" <<
e.what() << "'\n";
    return -1;
}

```

V. 결론 및 향후 연구방향

본 논문에서는 G-File 시스템을 이용한 GPIR 시스템 사용자가 시스템에 저장한 G-File의 속성을 관리하기 위한 알고리즘들을 제안하였다. GPIR 시스템은 사진 단말기(디지털카메라, 휴대폰 등)를 이용해 현재 위치(사진 촬영위치)를 알 수 있는 GPS, 사진 촬영 방향을 알 수 있는 자이로컴퍼스를 통해 서비스가 이루어진다. 이는 G-File 시스템을 이용하여 GPS 정

보와 사진 촬영 방향각을 사진파일에 포함하여 G-File을 생성하게 된다. G-File에 저장된 정보관리를 통하여 G-File 이용자는 자신의 목적에 맞는 파일을 관리하기 편리하게 되었고, 이는 많은 사용자가 G-File을 사용하여 사진파일을 관리할 수 있게 되었음을 의미한다.

본 알고리즘의 기대 및 파급효과로는 기존에 생성된 G-File의 관리가 손쉽게 이루어질 수 있게 될 수 있고, G-File 생성 기술을 활용하여 파생되는 새로운 기술 개발이 활성화 될 것으로 예상된다.

향후 연구로는 제안한 알고리즘의 손쉬운 버전 개발이 이루어지도록 해야 할 것이고, G-File의 관리가 보다 손쉽게 이루어질 수 있도록 관련 알고리즘을 개발하는 연구가 뒤따라야 할 것이다.

감사의 글

본 연구는 2011학년도 김포대학 연구비 지원에 의하여 연구되었음.

참고 문헌

- [1] 최상균, 이승재, 조동관, 김준영, 최재현, "데이터 베이스를 이용한 GPS 오차범위 최소화 시스템의 설계 및 구현", *한국엔터테인먼트산업학회 논문지* 제3권 제1호, 2009.3
- [2] 최상균, 김재생, "G-File을 이용한 GPIR 시스템 수신기 개발", *한국항공학회 논문지* 14권 3호, 2010.6
- [3] JEIDA, Digital Still Camera Image File Format Standard(Exchangeable image file format for Digital Still Cameras: Exif) Version 2.1, June 12, 1998
- [4] JIS X 4301:1995, <http://www.webstore.jsa.or.jp/>
- [5] 이현창, "너를 자극하는 C++ 프로그래밍", *한빛 미디어*, 2006.7
- [6] 이대호, 박영태, "영상처리를 위한 C++ 프로그래밍", *인터비전*, 2008.3

최 상 균 (崔常均)



1993년 8월 : 서강대학교 정보처리
학과(이학석사)

2005년 8월 : 경희대학교 컴퓨터
공학과(공학박사)

1998년 3월~현재 : 김포대학
e-비즈니스과 부교수

관심분야 : 컴포넌트 소프트웨어, 정보시스템 품질,
정보시스템 감리 등