

## IPTCメタデータを用いた職場巡視写真の処理： iPhotoと自作ツールを利用した例（第13回筑波大 学技術職員技術発表会報告集）

著者	北原 匡
雑誌名	筑波大学技術報告
号	34
ページ	42-47
発行年	2014-03
その他のタイトル	Processing of Workspace Inspections Photos by Using IPTC Metadata
URL	<a href="http://hdl.handle.net/2241/00123919">http://hdl.handle.net/2241/00123919</a>



アは多くはないが、Adobe Photoshop、Illustrator、Bridge、Apple Aperture などの他にフリーソフトも利用が可能である。本報告では、職場巡視写真の処理としてMacintosh 用の画像管理アプリケーションである iPhoto<sup>3</sup>を用い、IPTC メタデータを書き込んだ。

## 2.3 巡視写真におけるメタデータ

2.1 および 2.2 でメタデータについて述べたが、巡視写真の場合、メタデータを持つことのメリットとデメリットとして以下が考えられる。

巡視では一部屋あたり数枚撮影することが多い。巡視対象の部屋は約 800 あるが、一巡すると約 3000 枚となり、さらに経年の写真を保管しておかなければならない。これらの大量の画像を保存し管理するには、画像管理ソフトウェアや画像データベースを用いるのが便利であるが、巡視内容(日付、部屋番号、指摘の種類等)で検索するためには何らかのキーワードやタグ情報などのメタ情報が画像に紐付けされていなければならない。上記ソフトウェアではそのような仕掛けや構造を持っているので、そのソフトウェア内であれば問題はない。しかしながら、報告書を作成するなどで、ソフトウェアの外部に画像を取り出した時点で、その紐付けは無くなってしまいうため、何を撮影したものか判らなくなってしまう。単純に画像ファイル名に必要なメタ情報を付与する方法もあるが、ファイル名が変更されてしまう恐れがあるので適当ではない。

従って、図2に示すように、巡視で扱う画像ファイルの IPTC メタデータ領域に巡視情報を内包させることにより、画像ファイル単独でその写真の「意味」をも具備できるメリットがある。

一方で、メタデータを内包することによって、ファイルサイズが大きくなるデメリットもある。また、IPTC メタデータを扱うソフトウェアが必要である。

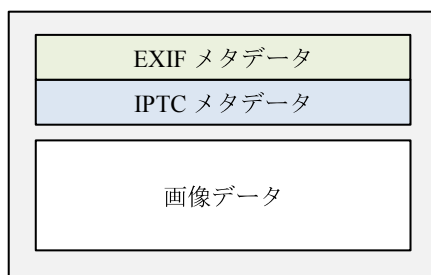


図2. メタデータを内包した JPEG ファイル

## 3. 巡視写真の処理ワークフロー

実際の巡視での写真撮影から報告書作成までの処理ワークフローをステップごとに述べる。撮影にはデジタルカメラを用い、画像処理には Macintosh 上の iPhoto と自作の報告書作成ツールを用いる。画像処理過程で、下記の IPTC メタデータを定義して利用する。

- ・ 部屋番号
- ・ 巡視日
- ・ 指摘項目
- ・ 指摘種別
- ・ 指摘の内容
- ・ 写真並びの順序番号

### 3.1 撮影

- ・ 撮影は全て 640 ピクセル× 480 ピクセルのランドスケープ(横長)とする。
- ・ 画像フォーマットは JPEG(EXIF)とする。
- ・ 指摘箇所の撮影に先立って、部屋番号プレートを撮影する。無い場合は適宜部屋番号が判るものを撮影する。

### 3.2 パソコンへの読み込みと整理

- ・ デジタルカメラと Macintosh を USB ケーブルで接続し、または SD カード等の記録メディアをマウントし、iPhoto で読み込む。
- ・ 読み込んだ複数枚の写真は、ライブラリ内に一つのイベントにまとめられて保存される。
- ・ 読み込んだイベントをクリックし、メニュー [表示：写真を並び替え：日付(昇順)] を指定する。イベント内の写真は撮影順に並ぶ。
- ・ 各画像の下には画像ファイル名が表示されている。これを図3のように部屋ごとに下記のような文字列に修正する。この項目は IPTC メタデータ(タイトル)として記録される。

3A100\_131225000

3A100 : 部屋番号  
131225: 巡視日 (yyymmdd の形式)  
000 : 同一部屋内の画像順序番号  
(最初はすべて 000 としておく)



図3. 写真下の画像タイトル欄に記述 (iPhoto)

### 3.3 写真の取捨選択と並び替え

- ・ 不要な写真は削除または非表示とする。

<sup>3</sup> Apple iPhoto '11 Ver.9.4.3 を用いた。

- ・ 巡視の内容にあわせて写真の並び順をドラッグして変更する。
- ・ 各画像の下のタイトル欄にシーケンス番号を付与する。(3.2 での記述に上書き)

3A100\_131225001, 3A100\_131225002, ...

- ・ メニュー [表示 : 写真を並び替え : タイトル (昇順)] を指定する。イベント内の写真はタイトル順に並ぶ。

### 3.4 巡視情報の書き込み

以下のような、写真が示す巡視情報を各画像の情報 (説明・キャプション) 欄に書き込む。

<転倒 : 新規>入って右の書庫が壁・床に固定されていません。

この文字列は 3.4.1 から 3.4.3 に述べる巡視情報の項目から構成されている。

#### 3.4.1 指摘項目

指摘の内容を 12 項目に分類し、それぞれに対応した下記の文字列の一つを記述する。

<整頓 : <転倒 : <落下 : <ボンベ :  
<配線 : <通路 : <試薬 : <標識 :  
<騒音 : <照度 : <Co2 : <特記 :

#### 3.4.2 指摘種別

今回の指摘のタイプを 3 種類に分類し、それぞれに対応した下記の文字列の一つを記述する。

新規> 継続> 改善>

#### 3.4.3 指摘の内容

指摘内容および改善策について簡潔に的確かつ具体的に記述する。(自由形式)

例)入って右の書庫が壁・床に固定されていません。

#### 3.4.4 情報欄に書き込む

iPhoto ウィンドウの右下の「情報」ボタンをクリックすると図 4 に示す情報欄 (説明・キャプション欄) が表示されるので、3.4.1 から 3.4.3 で生成した文字列を順に連結したものを書き込む。

### 3.5 画像の書出し

書き出す画像を全て選択し (非表示の×アイコンが付いている写真は処理対象外となる)、メニュー [ファイル : 書き出す...] を選択して表示されるダイアログで図 5 のように設定する。



図 4. 指摘内容を情報欄に記述した例 (iPhoto)

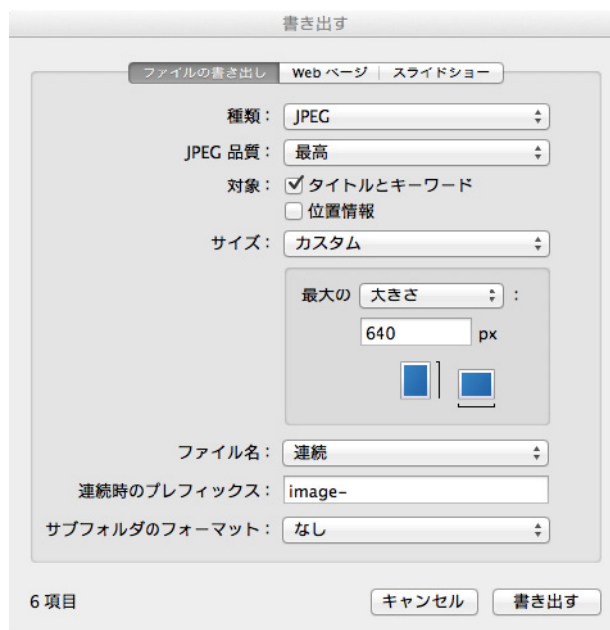


図 5. 写真を JPEG ファイルとして書き出す (iPhoto)

[書き出す] ボタンをクリックすると、書出し先を聞いてくるので、任意のフォルダを選択し、[OK] ボタンをクリックする。

### 3.6 画像ファイルの処理と報告書作成

画像が書き出されていることを確認し、処理アプリケーションツールアイコン (図 6) をダブルクリックして起動し、以下の手順で実行する。

- ・ 処理対象の写真を含むフォルダを選択する。
- ・ 今回の巡視日 (3.2 で記述した日) を yymmdd の形式で入力する。
- ・ 前回巡視した日を yymmdd の形式で入力する。(省略可能)



図 6. アプリケーションツールのアイコン



### 3.7 処理結果

3.1 から 3.6 までの処理結果を図 7 に示す。部屋番号名のフォルダ内に HTML 形式の index.html ファイルと画像ファイルによる写真報告書コンテンツが作成され、Web ブラウザで表示する。コンテンツ内の各画像ファイルは、IPTC メタデータの情報をもとに下記のようにリネームされている。

元画像ファイル名 : image-1.jpg  
処理後画像ファイル名 : 3A100\_131225001\_n\_te.jpg

上記の画像ファイル例では、そのファイル名から 20131225 に実施の 3A100 室の職場巡視において、転倒(te)に関しての新規(n)の指摘写真であることがわかる。万が一ファイル名が変更されても、自身の IPTC メタデータからファイル名は復元可能である。また、これらの画像ファイルを集めれば、別途管理ソフトウェアを使わなくとも、デスクトップ上でソートし、検索や管理することができる。

写真報告書は、HTML 形式のほかに PDF 形式でも同時に作成される。印刷する場合は、PDF を用いた方がレイアウトの崩れ等が生じないので便利である。

生成された写真報告書の表示例を図 8 に示す。

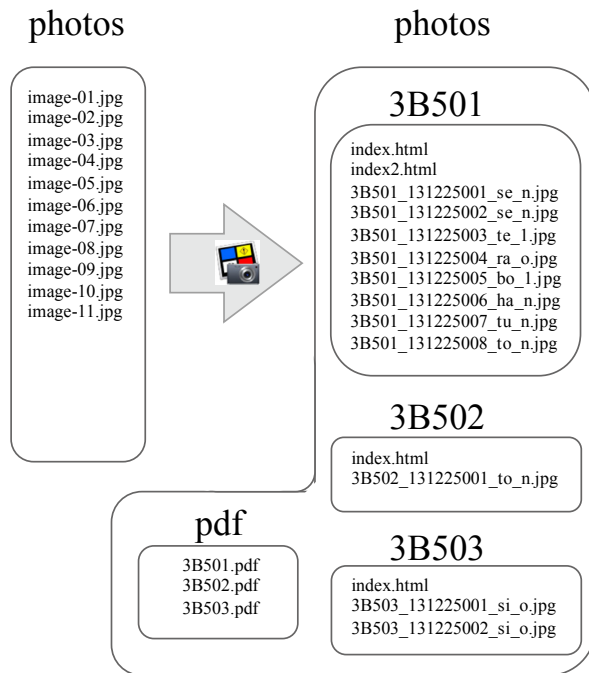


図 7. 処理ツール実行後のファイル構成

### 4. 自作の処理アプリケーションツール

写真報告書作成の処理で用いた自作のアプリケーションツールについて、主な仕様などを説明する。

本アプリは、Macintosh OS X の環境で、AppleScript、PHP、bash で記述されたスクリプトを Automator<sup>[1]</sup> (様々なスクリプトを組み合わせ、自動処理ワークフローを構築する OS X 標準のツール) 上で一つのアプリケーションとして作成したものであり、図 9 に示すフローで実行される。



図 8. 写真報告書の出力例

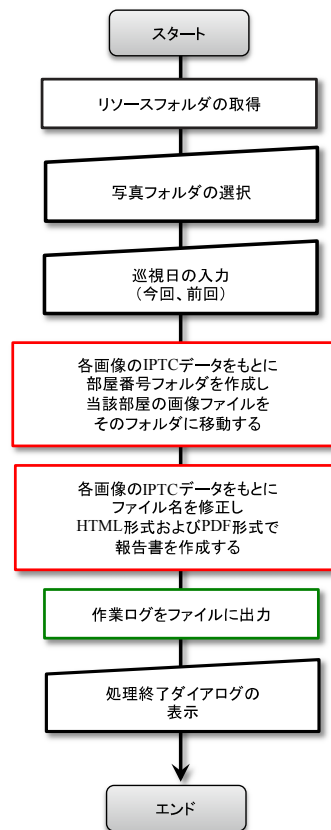


図 9. 自作アプリケーションの動作フロー

黒枠 : AppleScript で記述  
赤枠 : PHP で記述  
緑枠 : bash で記述

## 4.1 IPTC メタデータの取得

JPEG 画像内の IPTC メタデータを取得する部分は本アプリで最も重要であり、下記のような PHP スクリプトで実現している<sup>[2]</sup>。スクリプト内の APP13 は、IPTC メタデータ領域を示すマーカーである。

```
<?php
    $size = GetImageSize ("filename.jpg", $info);
    $iptc = iptcparse ($info["APP13"]);
    if(isset($info["APP13"])) {
        if(is_array($iptc)) {
            $graphic_name = $iptc["2#005"][0];
            $caption = $iptc["2#120"][0];
        }
    }
?>
```

このスクリプト例では、画像ファイル *filename.jpg* の IPTC メタデータから、`$graphic_name` 変数に 3.3 の情報が、`$caption` 変数に 3.4 の情報がそれぞれ格納される。

## 4.2 HTML ファイルの出力

写真報告書は基本的に HTML ファイルで作成され、ブラウザで閲覧する。HTML ファイルはテキストファイルであるので、コーディング上特に問題はないが、写真報告書のレイアウトを決定する部分であるので、CSS (スタイルシート) の細かい調整や Web 標準に適合させるための書式を調整する必要があった。

また、報告書 1 ページには最大 6 枚の画像を表示するが、それ以上の枚数の場合のページ追加処理が必要であった。

## 4.3 PDF ファイルの出力

写真報告書の印刷には、HTML 形式より PDF 形式のほうが適している。そのために、4.2 で出力された HTML 形式のものを PDF に変換する機能をもつフリーの PHP クラスである mPDF<sup>[3]</sup> を用いた。例として *index.html* ファイルを *index.pdf* に変換するコードを以下に示す。

与えられた HTML ファイルのブラウザ結果と同じレイアウトを得るためには、HTML ファイル内の CSS を微調整する必要がある。

```
<?php
    include("mpdf/mpdf.php");
    $html = file_get_contents("index.html");
    $mpdf = new mPDF('ja', 'A4');
    $mpdf->mirrorMargins = true;
    $mpdf->setDisplayMode('fullpage');
    $mpdf->writeHTML($html);
    $mpdf->Output("index.pdf", 'F');
?>
```

また、複数の PDF を一つの PDF にまとめるために PDFMerger<sup>[4]</sup> を用いた。下記は *pdf1.pdf* と *pdf2.pdf* を結合して *pdf3.pdf* として出力するコード例である。

```
<?php
    include("PDFMerger/PDFMerger.php");
    $pdf = new PDFMerger;
    $pdf->addPDF("pdf1.pdf", 'all')
        ->addPDF("pdf2.pdf", 'all')
        ->merge('file', "pdf3.pdf");
?>
```

上記の PHP クラスは、アプリケーションツールパッケージ内 (ツール名 *app/Contents/Library/*) に配置し、PHP スクリプトからインクルードする。

## 5. まとめと課題

画像ファイルに内包されるメタデータに情報を持たせることにより、自己説明型の画像となり、そのメタデータを用いて様々な後処理が可能である。職場巡視で撮影された写真の管理にも有用である。また、アプリケーションツールにより、簡単に写真報告書を作成することができ、実際の業務で有益に使用している。

一方で、一連のワークフローでは IPTC メタデータを扱うことができる画像管理ソフトウェアが必要である点、および、作成したアプリケーションツールが Macintosh OS X 環境でしか使えない点で、必ずしも汎用的とは言えない。とはいえ、PHP スクリプトが実行できる環境 (例えば Unix、PHP 環境をインストールした Windows 等) があれば、コマンドベースでスクリプト単体で実行は可能であり、何らかの IPTC メタデータの書き込みができるソフトウェアさえあれば、同様のワークフローで作業することができる。

## 参考文献

- [1] Automator;  
<http://macwiki.sourceforge.jp/wiki/index.php/Automator>
- [2] iptcparse;  
<http://www.php.net/manual/ja/function.iptcparse.php>
- [3] mPDF;  
<http://mpdf.bpm1.com/>
- [4] PDFMerger;  
<http://pdfmerger.codeplex.com>

## Processing of Workspace Inspections Photos by Using IPTC Metadata

Masashi Kitahara

Technical Service Office for Systems and Information Engineering, University of Tsukuba,  
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan

Introducing the workflow for creating a report by utilizing the metadata area of the JPEG images for the processing stages of images taken during workspace inspection.

This workflow is structured around iPhoto image processing applications and newly developed application tools for Macintosh OS X. As well as being able to distinguish content of workspace inspection from just the image files, photo reports can also be easily created.

**Keywords:** Workspace Inspections, Photo, IPTC, Application