

Automatically reading GPS information from featured image and adding as post meta in WordPress

Wordpress has a semi-official way of geotagging, so I decided to fill this automatically from the featured image.

As for getting GPS information into the images, I used the following:

- [GPSLogger](#) on my phone; this can be set to create GPX, KML and txt files, so parsing them shouldn't be a problem
- [GPicSync](#) to read the according, closest entry from the GPX file and write it to my photos

This way I didn't have to buy an expensive GPS unit for my camera.

Unfortunately the solution requires:

- PHP `exec` function to be available
- `exiftool` to be installed in `$PATH`

I could not find any pure PHP implementation that could read GPS information from EXIF reliably.

```
add_action( 'init', 'extra_exif_init' );

function extra_exif_init() {
    add_filter( 'wp_read_image_metadata', 'read_extra_exif',
1, 3 );
}

function read_extra_exif ( $meta, $filepath = '',
$sourceImageType = ' ' ) {

    if (empty($filepath) || !is_file($filepath) || !
```

php

```

is_readable($filepath)) {
    // "{$filepath} doesn't exist"
    return $meta;
}

if ( $sourceImageType != IMAGETYPE_JPEG ) {
    // not JPEG means no EXIF
    return $meta;
}

$extra = array (
    'geo_latitude' => 'GPSLatitude',
    'geo_longitude' => 'GPSLongitude',
    'geo_altitude' => 'GPSAltitude',
    // feel free to add elements here, eg. 'lens' =>
'LensID'
);

$rextra = array_flip($extra);

$args = $metaextra = array();

foreach ($extra as $metaid => $exiftoolID ) {
    if (!isset($meta[ $metaid ])) {
        $args[] = $exiftoolID;
    }
}

if (!empty($args)) {
    $cmd = 'exiftool -s -' . join(' -', $args) . ' ' .
$filepath;
    // "Extracting extra EXIF for {$filepath} with
command {$cmd}"

    exec( $cmd, $exif, $retval);

    if ($retval == 0 ) {
        foreach ( $exif as $cntr => $data ) {
            $data = explode ( ' : ', $data );
            $data = array_map('trim', $data);
            if ( $data[0] == 'GPSLatitude' || $data[0] ==

```

```

'GPSLongitude' )
        $data[1] = exif_gps2dec( $data[1] );
        elseif ( $data[0] == 'GPSAltitude' )
            $data[1] = exif_gps2alt( $data[1] );

        $metaextra[ $rextra[ $data[0] ] ] = $data[1];
    }
}
}

$meta = array_merge($meta, $metaextra);

return $meta;
}

function exif_gps2dec ( $string ) {
    //103 deg 20' 38.33" E
    preg_match( "/([0-9.]+)\s?+deg\s?+([0-9.]+)'\s?+([0-9.]+)
\s?\s?+([NEWS])/", trim($string), $matches );

    $dd = $matches[1] + ( ( ( $matches[2] * 60 ) +
( $matches[3] ) ) / 3600 );
    if ( $matches[4] == "S" || $matches[4] == "W" )
        $dd = $dd * -1;
    return round($dd,6);
}

function exif_gps2alt ( $string ) {
    //2062.6 m Above Sea Level
    preg_match( "/([0-9.]+)\s?+m/", trim($string), $matches );

    $alt = $matches[1];
    if ( striistr( $string, 'below' ) )
        $alt = $alt * -1;
    return $alt;
}

```

Now, to add the actual meta to the post, I'll extend my previous auto-tagger^[4]. Just add the following inside the `autotag_by_photo` function found there:

```
// GPS
$try = array ( 'geo_latitude', 'geo_longitude',
'geo_altitude' );
foreach ( $try as $kw ) {
    $curr = get_post_meta ( $post->ID, $kw, true );
    // "Current {$kw} for {$post->ID} is: ${curr}"

    if ( isset ( $meta['image_meta'][$kw ] ) && !
empty( $meta['image_meta'][$kw ] ) ) {
        if ( empty ( $curr ) ) {
            add_post_meta( $post->ID, $kw, $meta['image_meta'
][ $kw ], true );
        }
        elseif ( $curr != $meta['image_meta'][$kw ] ) {
            update_post_meta( $post->ID, $kw,
$meta['image_meta'][$kw ], $curr );
        }
    }
}
```

Links

1. <https://fossdroid.com/a/gpslogger.html>
2. <https://github.com/metadirective/GPicSync>
3. <http://owl.phy.queensu.ca/~phil/exiftool/>
4. <https://petermolnar.net/wordpress-automate-content-featured-image-iptc/>

Created by Peter Molnar <mail@petermolnar.net>, published at 2016-06-22 20:00 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/gps-from-featured-wordpress/> , licensed under CC-BY-4.0 .