

## Xavier Lacot

- [Go to the content](#)
- [Go to the menu](#)

## Home / Géolocalisez vos photos en PHP

- [geek](#)
- [private](#)

**Cet article est la reprise d'un article publié dans le magazine [PHP Solutions](#) de Mars 2007, que vous trouverez encore en kiosque si vous vous pressez un peu.**

Avec l'avènement de services de cartographie avancés sur le Web, comme [Google Maps](#) ou [Yahoo! Maps](#), la localisation géographique de données peut désormais aisément être mise en œuvre par les concepteurs Web. Cet article présente un processus de géolocalisation de photographies, depuis la collecte des données jusqu'à leur affichage sur une carte Google Maps, en passant par la synchronisation entre le GPS et l'appareil photo et le marquage [EXIF](#) [1] des photos.

### Sommaire

1. [Qu'est-ce que la géolocalisation ?](#)
  1. [Présentation du concept de "géolocalisation"](#)
  2. [Géolocaliser, pourquoi ?](#)
2. [Du terrain aux cartes : le processus de géolocalisation passé en revue](#)
3. [Récolte des données](#)
4. [Import et analyse des données](#)
  1. [Calcul de la position de prise de vue d'une photo](#)
  2. [Marquage EXIF des images](#)
5. [Présentation des photos géolocalisées sur le Web](#)
  1. [Présentation sous forme de flux sémantique](#)
6. [Conclusion : vers Google Maps...](#)
7. [Sur Internet](#)
8. [Terminologie](#)

### Qu'est-ce que la géolocalisation ?

#### Présentation du concept de "géolocalisation"

Comme son nom l'indique, la géolocalisation est la localisation géographique, sur une carte, d'un objet ou d'une personne. La mouvance du "Web 2.0" propose déjà de nombreuses applications présentant des données géographiques, les plus éminentes étant bien évidemment Google Maps et Yahoo! Maps. Depuis quelques temps, le mouvement s'est accéléré : début Septembre 2006, [Flickr](#) lance un service de géolocalisation des photographies ajoutées par ses membres, tandis que nombreux "mashups" [2] apparaissent, et exploitent l'API proposée par Google Maps pour présenter leurs données.

Pour autant, les problématiques de la localisation ne sont pas récentes. En vérité, les premiers systèmes de positionnement sont apprus dans les années 40, avec notamment le système de radio-navigation [LORAN](#). Plusieurs système de navigation ont depuis été mis au point, le plus répandu étant [GPS](#) [3]. C'est à l'aide de signaux GPS que, dans cet article, nous allons effectuer la géolocalisation de photographies.

#### Géolocaliser, pourquoi ?

La géolocalisation, c'est donc la capacité à situer un objet à un instant donné, par le biais de ses coordonnées, de manière à pouvoir ensuite le localiser sur une carte. Les applications sont multiples : retrouver un skieur perdu en montagne, marquer un beau point de vue pour y revenir plus tard, ou tout simplement associer géolocalisation et cartographie pour pouvoir se déplacer plus aisément – c'est ce que font les GPS de navigation automobile.

Dans le cas du Web, la géolocalisation de photographies répond à des objectifs différents : l'envie de communiquer, d'associer des images à des lieux, de créer une vision collective du monde, partagée à l'échelle de l'humanité. Géolocaliser ses photographies peut avoir des aspects plus pratiques : créer des visites virtuelles, répondre à des besoins encyclopédiques ou journalistiques, etc. D'une manière générale, une photographie géolocalisée n'est plus une simple image : elle est le témoignage de l'état du monde en un instant et un lieu donnés.

#### Du terrain aux cartes : le processus de géolocalisation passé en revue

Contrairement à Flickr, qui propose d'effectuer la géolocalisation des images "en ligne" par le biais d'une carte sur laquelle l'utilisateur choisit l'endroit d'où une photo a été prise, j'ai pensé qu'il était intéressant d'automatiser le processus, de sorte que mon intervention dans la géolocalisation soit aussi minimale que possible. Mettre sur pieds une galerie photographique en ligne représente un travail assez fastidieux dans la gestion des données, la retouche éventuelle des images, les commentaires, etc., de sorte qu'il m'a semblé peu nécessaire d'ajouter encore du travail à cette tâche déjà assez prenante. Le procédé que j'ai mis en place sur mon site personnel pour géolocaliser mes photographies se décompose donc en plusieurs étapes successives :

1. la collecte des données, c'est-à-dire les prises de vue et l'enregistrement de la position géographique à l'aide d'un GPS,
2. l'import des données depuis l'appareil photo et le GPS, sous [Ubuntu Linux](#),
3. le marquage des photographies avec les données de géolocalisation,
4. la production d'un flux sémantique indiquant la position des images présentées sur le site,
5. l'utilisation de ce flux pour afficher les images sur une carte Google Maps.

## Récolte des données

Une chose est certaine : même si on a un GPS à disposition, il n'est pas pratique de devoir appuyer sur un bouton pour enregistrer sa position à chaque prise de vue. Avant de commencer ce projet, j'ai donc choisi de m'équiper d'un GPS capable d'enregistrer des traces, c'est-à-dire des séries de points enregistrées en continu et représentant ma position horodatée. Il suffit donc uniquement d'allumer le GPS en début de promenade photo, le mettre dans son sac, pour ensuite pouvoir retrouver les coordonnées correspondant à une certaine heure de la journée.

Parallèlement à ça, tous les appareils numériques ont, par chance, la capacité de produire des fichiers JPEG marqués avec des "informations EXIF". Ces informations EXIF sont assez variées : type d'appareil, conditions de prise de vue (ouverture, vitesse d'obturation, déclenchement du flash, etc.), et surtout date et heure de prise de vue. A partir de ces deux derniers éléments, et à l'aide de la trace enregistrée par le GPS, il est possible d'induire avec une précision assez intéressante les coordonnées de l'endroit d'où une photo a été prise.

Une séance de photographie avec GPS doit donc systématiquement débiter par une synchronisation des horloges des appareils. Un décalage d'une minute peut produire des erreurs de retranscription assez importantes – de l'ordre de plusieurs dizaines de mètres – alors qu'en terrain dégagé, un bon GPS peut avoir une précision de quelques mètres seulement.

## Import et analyse des données



La première problématique concerne l'import des données depuis le GPS et l'appareil photo sous Linux. Si, pour les appareils photo, il existe toujours un mode "clé USB" (périphérique de masse USB), la diversité du marché des GPS impose une analyse assez fine des différents produits. Si le GPS employé ne propose pas un moyen simple d'accéder aux traces enregistrées sous Linux, tout le

processus tombe à l'eau. Outre la connectivité à un ordinateur, d'autres éléments à prendre en compte au moment de l'achat d'un GPS, il faut veiller à :

- l'autonomie de l'appareil, son mode d'alimentation. Une alimentation par piles est plus coûteuse qu'une batterie rechargeable sur le long terme, mais elle permet une plus grande autonomie lorsqu'on est éloigné de toute source d'énergie.
- la sensibilité à la réception des signaux GPS : un GPS est un récepteur de signaux radio et, comme il existe sur le marché des téléphones portables qui offrent une bonne réception et d'autres qui sont plus sensibles aux bâtiments, la qualité de réception varie vraiment d'un GPS à un autre. Les GPS sont généralement proposés sous forme de gammes dédiées à certains usages, il convient de choisir un GPS dédié au trekking ou à la marche en milieu peu dégagé.
- la capacité d'enregistrement de l'appareil, qui conditionne la qualité des traces enregistrées par le GPS, et le nombre de points pour chaque trace. Certains GPS ne peuvent enregistrer que mille points par trace, ce qui représente à peine trois heures à raison d'un point toutes les dix secondes.
- le format d'enregistrement des traces. Même si certains logiciels open-source permettent de manipuler les différents formats de trace du marché, il est toujours bon de disposer d'un format directement exploitable.

J'ai choisi le modèle [GPSMap 60Cx](#) de [Garmin](#), car il correspondait à tous ces besoins. Il propose un mode "clé USB" et fournit des traces de 10000 points, au format XML GPX (voir listing 1).

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<gpx>
<trk>
<name>ACTIVE LOG150402</name>
<trkseg>
<trkpt lat="52.366632" lon="4.891696">
<ele>8.409</ele>
<time>2006-09-19T13:04:02Z</time>
</trkpt>
<trkpt lat="52.366834" lon="4.891727">
<ele>8.644</ele>
<time>2006-09-19T13:04:12Z</time>
</trkpt>
<trkpt lat="52.366801" lon="4.891744">
<ele>8.677</ele>
<time>2006-09-19T13:05:11Z</time>
</trkpt>
<trkpt lat="52.366790" lon="4.891735">
<ele>6.984</ele>
<time>2006-09-19T13:05:28Z</time>
</trkpt>
</trkseg>
</trk>
</gpx>
```

Disposer d'un flux au format XML est un gros avantage ; cela signifie entre autre que l'on va pouvoir employer [SimpleXML](#) pour lire ces données avec PHP. Ceci dit, certains logiciels comme [gpsbabel](#), permettent de réaliser aisément des conversions entre formats de données GPS, ce qui permet donc d'obtenir facilement un flux semblable à celui présenté dans le listing 1.

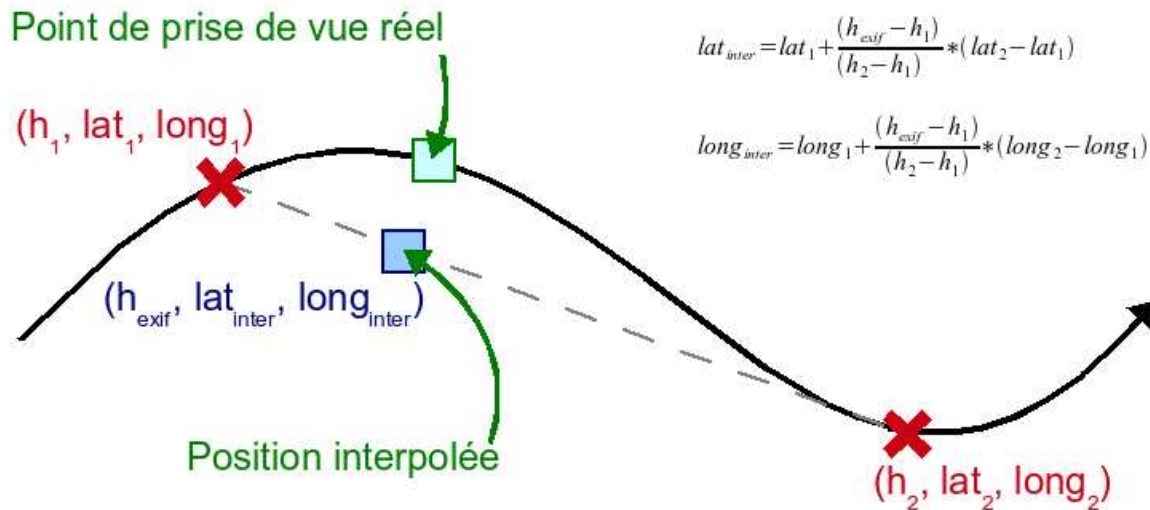
Le code présenté sur la listing 2 indique l'import des données d'un fichier de trace au format GPX dans une base de données. La syntaxe employée correspond au modèle proposé par [Propel](#), l'ORM livré avec le framework [Symfony](#), utilisé pour ma galerie photo :

```
<?php
$trace_dom = simplexml_load_file('trace.gpx');

foreach ( $trace_dom->trk as $track )
{
    foreach ( $track->trkseg as $segment )
    {
        foreach ( $segment->trkpt as $point )
        {
            $tracepoint = new TracePoint();
            $tracepoint->setDate($point->time);
            $tracepoint->setLatitude($point['lat']);
            $tracepoint->setLongitude($point['lon']);
            $tracepoint->setAltitude($point->ele);
            $tracepoint->save();
        }
    }
}
?>
```

### Calcul de la position de prise de vue d'une photo

Une fois que les traces GPS sont disponibles sous forme de points en base de données, il est possible de marquer les photographies, c'est-à-dire de leur associer des coordonnées géographiques. Comme expliqué précédemment, la trace enregistrée par le GPS n'est qu'une succession de points horodatés : il est donc tout à fait possible qu'une photo ait été prise au cours de l'intervalle de temps séparant deux points. Dans ce cas, il est nécessaire d'interpoler la position de prise de vue, située entre les deux points :



Cette interpolation ne doit être effectuée que si les deux points ne sont pas trop éloignés temporellement de l'instant de prise de vue. Si l'on souhaite obtenir une position interpolée de précision convenable, il est nécessaire que les deux points qui l'encadrent dans le temps ne soient pas trop éloignés. Un écart de 20 secondes convient assez bien en règle générale, mais cela dépend de votre mode et de votre vitesse de déplacement.

Le listing 3 présente la méthode adoptée pour interpoler la position géographique de prise de vue d'un cliché.

```
<?php
$image_exif_data = exif_read_data('image.jpg');

if ( isset($image_exif_data['DateTimeOriginal']) )
{
    $image_date = $image_exif_data['DateTimeOriginal'];
    $image_date = str_replace(' ', ':', $image_date);
    $image_date = explode(':', $image_date);
    $image_date = mktime($image_date[3], $image_date[4], $image_date[5], $image_date[1], $image_date[2], $image_date[0]);

    // get image location from trace database :
    // 1- get point after, within a 10 seconds window
    // 2- get point before, within a 10 seconds window
    // 3- if close points found, compute image location with a proportional
    // rule
    $c = new Criteria();
    $criterion = $c->getNewCriterion(TracePointPeer::DATE, $image_date, Criteria::GREATER_EQUAL);
    $criterion->addAnd($c->getNewCriterion(TracePointPeer::DATE, $image_date + 10, Criteria::LESS_THAN));
    $c->add($criterion);
    $c->addAscendingOrderByColumn(TracePointPeer::DATE);
    $c->setLimit(1);
    $point_after = TracePointPeer::doSelect($c);

    $c = new Criteria();
    $criterion = $c->getNewCriterion(TracePointPeer::DATE, $image_date, Criteria::LESS_EQUAL);
    $criterion->addAnd($c->getNewCriterion(TracePointPeer::DATE, $image_date - 10, Criteria::GREATER_THAN));
    $c->add($criterion);
    $c->addDescendingOrderByColumn(TracePointPeer::DATE);
    $c->setLimit(1);
    $point_before = TracePointPeer::doSelect($c);

    if ( isset($point_after[0]) )
    {
        if ( !isset($point_before[0]) || $point_after[0]->getDate() == $image_date )
        {
            $image_latitude = $point_after[0]->getLatitude();
            $image_longitude = $point_after[0]->getLongitude();
            $image_altitude = $point_after[0]->getAltitude();
        }
        else
        {
            $coefficient = ($image_date - strtotime($point_before[0]->getDate())) / (strtotime($point_after[0]->getDate()) - strtotime($point_before[0]->getDate()));
            $image_latitude = $point_before[0]->getLatitude() + $coefficient * ($point_after[0]->getLatitude() - $point_before[0]->getLatitude());
            $image_longitude = $point_before[0]->getLongitude() + $coefficient * ($point_after[0]->getLongitude() - $point_before[0]->getLongitude());
            $image_altitude = $point_before[0]->getAltitude() + $coefficient * ($point_after[0]->getAltitude() - $point_before[0]->getAltitude());
        }
    }
    else if ( isset($point_before[0]) )
    {
        $image_latitude = $point_before[0]->getLatitude();
        $image_longitude = $point_before[0]->getLongitude();
        $image_altitude = $point_before[0]->getAltitude();
    }
}
```

```
}
?>
```

## Marquage EXIF des images

Le listing 3 permet d'obtenir la position de prise de vue de la photographie sous forme de trois variables (\$image\_latitude, \$image\_longitude, \$image\_altitude), que l'on peut maintenant librement exploiter : enregistrement en base de données, marquage EXIF des images, etc.

PHP propose par défaut des fonctions de lecture des métadonnées EXIF, notamment avec `exif_read_data()`. Cependant, il n'est pas possible d'écrire aisément des champs EXIF au sein d'une image avec PHP. Fort heureusement, le projet [PHP Exif Library](http://phpexif.org/) permet de manipuler les données EXIF contenues au sein d'une image. PEL est écrit en PHP5, et a l'avantage de permettre l'édition de tous les champs EXIF, contrairement à certaines autres bibliothèques qui ne se bornent généralement qu'aux champs EXIF les plus courants.

La documentation de PEL est très complète et la liste de diffusion du projet assez vivante, de sorte qu'il est aisé de trouver des exemples complets d'utilisation de cette bibliothèque. C'est particulièrement pratique pour les assez marginales utilisations de PEL, comme le geotagging de photographies, par exemple. La documentation de l'API de PEL indique en détail le format des informations qui peuvent être ajoutées aux tags EXIF (cf. par exemple [http://pel.sourceforge.net/doc/PEL/PelTag.html#GPS\\_ALTITUDE](http://pel.sourceforge.net/doc/PEL/PelTag.html#GPS_ALTITUDE)).

Les coordonnées (latitude et longitude) doivent être écrites sous la forme de tableaux associatifs représentant les degrés, minutes et secondes. Le listing 4 propose une méthode permettant d'effectuer cette conversion, pour obtenir le format attendu par PEL.

```
<?php
/**
 * Convert a decimal value to a degree/minute/second value
 *
 * @param $decimalvalue decimal value
 * @return two dimensionnal array
 */
private function convertDecimalToDMS($decimalvalue)
{
    $degree = array(floor($decimalvalue), 1);
    $rest = 60 * ($decimalvalue - floor($decimalvalue));
    $minutes = array(floor($rest), 1);
    $rest = 60000 * ($rest - floor($rest));
    $seconds = array(floor($rest), 1000);
    return array($degree, $minutes, $seconds);
}
?>
```

Cette méthode peut être directement employée pour effectuer le marquage EXIF des photographies ; PEL est un peu verbeux dans son utilisation, aussi le listing 5 montre-t-il comment faire en détail.

```
<?php
if ( isset($image_latitude) && isset($image_longitude) )
{
    // get the GPS EXIF section
    $jpeg = new PelJpeg();
    $jpeg->loadFile($picture_file);
    $exif = $jpeg->getSection(PelJpegMarker::APP1);
    $tiff = $exif->getTiff();
    $ifd0 = $tiff->getIfd();
    $gps = $ifd0->getSubIfd(PelIfd::GPS);

    if ( !$gps )
    {
        $gps = new PelIfd(PelIfd::GPS);
        $ifd0->addSubIfd($gps);
    }

    // a point has been found => set this data in the image,
    $latitude = $this->convertDecimalToDMS($image_latitude);
    $lat = new PelEntryRational(PelTag::GPS_LATITUDE, $latitude[0], $latitude[1], $latitude[2]);
    $gps->addEntry($lat);
    $latref = new PelEntryAscii(PelTag::GPS_LATITUDE_REF, ($image_latitude > 0) ? 'N' : 'S');
    $gps->addEntry($latref);

    $longitude = $this->convertDecimalToDMS($image_longitude);
    $long = new PelEntryRational(PelTag::GPS_LONGITUDE, $longitude[0], $longitude[1], $longitude[2]);
    $gps->addEntry($long);
    $longref = new PelEntryAscii(PelTag::GPS_LONGITUDE_REF, ($image_longitude > 0) ? 'W' : 'E');
    $gps->addEntry($longref);

    $altitude = abs(floor($image_altitude * 1000)).'/1000';
    $alt = new PelEntryRational(PelTag::GPS_ALTITUDE, $altitude);
    $gps->addEntry($alt);
    $altref = new PelEntryByte(PelTag::GPS_ALTITUDE_REF, ($image_altitude >= 0) ? '0' : '1');
    $gps->addEntry($altref);

    // save the geotagged file
```

```
file_put_contents('image.jpg', $jpeg->getBytes());
}
?>
```

## Présentation des photos géolocalisées sur le Web

La méthode de géolocalisation présentée dans la partie précédente peut tout à fait être intégrée au module de publication d'un site Web. L'idée, naturellement, est de rendre ensuite ces informations accessibles aux utilisateurs – et ne pas se contenter uniquement de les ajouter aux données EXIF des photographies taguées.

L'extraction des informations EXIF GPS fonctionne parfaitement bien avec `exif_read_data()`. Ceci dit, les champs `GPSLatitude` et `GPSLongitude` lus par PHP se présentent sous forme de tableaux associatifs ; il est donc nécessaire d'effectuer la transformation inverse à celle effectuée par la méthode `convertDecimalToDMS()` présentée plus haut.

## Présentation sous forme de flux sémantique

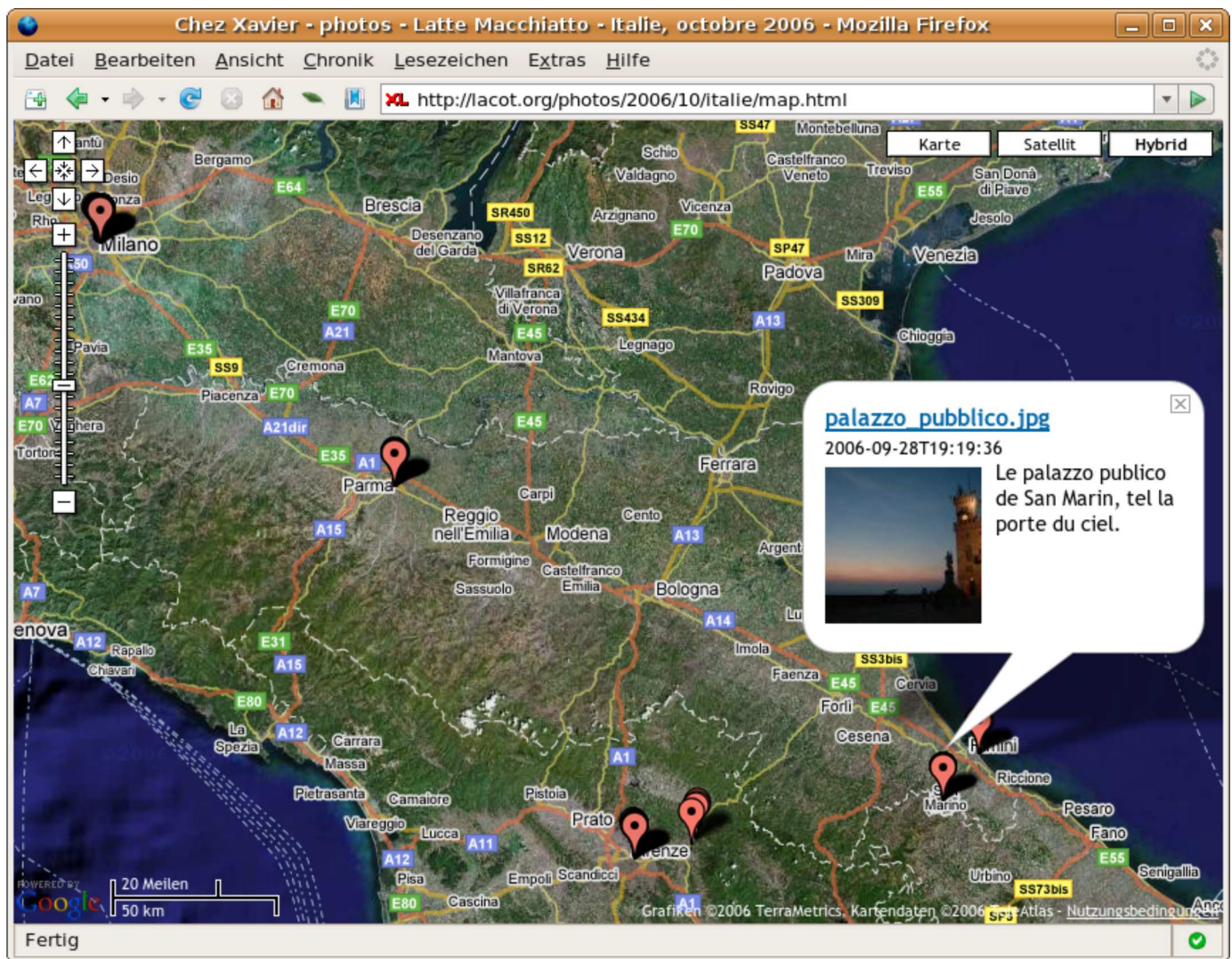
Une galerie d'images peut aisément se présenter sous forme d'un flux sémantique. Plusieurs méthodes sont possibles, de l'assemblage de vocabulaires [RDF](#) variés au choix d'un langage d'ontologies existant. En Février 2006, Opéra introduisait, sur son blog dédié au Web sémantique [\[4\]](#), un [schéma adapté à la représentation des galeries de photographies](#), qui répond parfaitement aux besoins de notre exemple.

Cependant, le modèle proposé par Opéra ne permet pas d'inclure des données géographiques au sein du flux RDF. Pour cela, il faut faire appel à d'autres vocabulaires RDF : les deux solutions les plus séduisantes sont [GeoRss](#) ou un vocabulaire créé par le W3C, [WGS84](#). C'est ce dernier vocabulaire qui est employé dans le listing 6, où on peut voir un exemple de présentation sémantique d'une galerie géolocalisée. Les coordonnées (`geo:lat` et `geo:long`) doivent être des valeurs décimales, et l'altitude un nombre indiquant l'élévation en mètres du lieu que l'on veut indiquer.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:gal="http://my.opera.com/community/xmlns/2006/gallery#"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
xmlns:foaf="http://xmlns.com/foaf/0.1/">>
<gal:Gallery rdf:about="http://lacot.org/photos/2006/10/italie.rdf">
<foaf:maker rdf:resource="http://lacot.org/public/xavier-lacot-foaf.rdf#XavierLacot"/>
<gal:image>
<foaf:Image rdf:about="http://lacot.org/photos/2006/10/italie/palazzo_pubblico.jpg">
<rdfs:label xml:lang="fr">palazzo_pubblico.jpg</rdfs:label>
<dc:description xml:lang="fr">Le palazzo publico de San Marin, tel la porte du ciel.</dc:description>
<dc:date>2006-09-28T19:19:36</dc:date>
<foaf:thumbnail rdf:resource="http://lacot.org/cache/photos/front/italie-1381.jpg"/>
<foaf:homepage rdf:resource="http://lacot.org/photos/2006/10/italie/1381.html"/>
<geo:lat>43.9364</geo:lat>
<geo:long>12.447</geo:long>
</foaf:Image>
</gal:image>
</gal:Gallery>
</rdf:RDF>
```

## Conclusion : vers Google Maps...





Nous atteignons pas à pas à l'objectif final : la présentation des photographies sur une carte. Pour réaliser cela, il faut d'abord s'enregistrer auprès de Google pour obtenir une clé autorisant à employer son service de cartographie. Ensuite, une solution intéressante consiste à faire appel en Javascript au flux RDF précédemment généré, le parser, et afficher un point sur la carte par image géolocalisée.

On peut agrémenter l'affichage en groupant plusieurs photos, ou en affichant une boîte d'affichage détaillé au survol d'un point. La documentation de l'API de Google Maps est très complète, il suffit de laisser libre court à son imagination ! Une [démonstration du résultat final](#) est disponible en ligne.

On peut décider d'ajouter d'autres métadonnées aux images : direction de prise de vue, angle utilisé, etc. Dans tous les cas, la mise en place d'un processus de gestion de l'information, et la mise à disposition de celle-ci sous la forme de flux sémantique sont essentiels, car ils permettent ensuite de visualiser cette information sous des formes très diverses.

## Sur Internet

- <http://www.geospatialsemanticweb.com/> – présente les dernières avancées du Web Sémantique en terme de géolocalisation,
- <http://www.google.com/apis/maps/> – présente l'API de Google Maps,
- [http://lacot.org/trac/file/tags/0.5-milano/lib/lacot\\_org/ImageHandler.class.php](http://lacot.org/trac/file/tags/0.5-milano/lib/lacot_org/ImageHandler.class.php) – le code source présenté dans ce tutoriel.

## Terminologie

- [1] EXIF est l'acronyme de "Exchangeable image file format", et désigne une spécification permettant d'intégrer des métadonnées au sein de fichiers numériques. EXIF est supporté par les formats de fichier JPEG et TIFF.
- [2] "Mashup" est un terme qui désigne une application Web composite, c'est-à-dire une application qui assemble les ressources et les données fournies par d'autres sites Web afin de fournir un contenu plus riche.
- [3] GPS ou "Global Positioning System" est le principal système de positionnement mondial actuellement disponible. L'avènement du système Galileo devrait permettre d'augmenter la précision des appareils de géolocalisation, actuellement bornée à plusieurs mètres avec GPS.
- [4] Le Web sémantique est constitué d'un ensemble de technologies et de langages Web destinés à ajouter une sémantique formelle au Web, de sorte que les machines puissent, elles aussi, en interpréter le contenu.
- written on: 2007-04-09

## Comments

There is no comment for the moment. [Add a new comment](#)

- [← said before...](#)
- [said after... →](#)
- [how ? Writings, photos](#), etc.
- [what ?](#) You said tags ?
- [when ?](#) Some dates count.
- [who ?](#) A sort of *eponimia*.
- [about](#) here
- [en](#)