AUTOMATIC REGISTRATION OF VECTOR DATA WITH OPTICAL IMAGES

Y. Tanguy[1]*, J. Michel[1], G. Salgues[2]

[1] Centre National d'Etudes Spatiales (CNES), 18 avenue E. Belin, Toulouse cedex 9, France
[2] Magellium, 1, rue Ariane, 31520 Ramonville Saint Agne

Commission IV, WG IV/3

KEY WORDS: Optical satellite imagery, vector-to-image registration, vector conflation, feature extraction, Orfeo ToolBox

ABSTRACT:

This paper presents a method to perform automatic vector-to-image registration. The proposed method performs well on different kinds of optical satellite images from Very High Resolution (VHR, sub-meter resolution) to images in the 10/20m resolution range. It allows to automatically register vector dataset such as urban maps (by using building layers). In contrast with existing methods, our method needs few prior-knowledge on the features to match and can therefore adapt to different landscapes.
This paper demonstrates the method robustness in several use-cases and presents the implementation which will soon be available as open-source software.

1. CONTEXT

1.1 Introduction

Many geospatial applications need to combine large vector datasets with high resolution remote sensing imagery. Mapping techniques need a correct registration of remotely sensed images to achieve good performance and thus provide valuable georeferenced products.

Map to image registration main concern is to match a georeferenced database with a satellite image. This supposes to have an accurate georeferenced image, and then to match some features in the image with their counterparts in the map. Those matched features are then used to estimate residual alignments that should be corrected, whether by shifting the vector dataset (OSM, 2015) or applying the inverse transform on the image.

Manual map correction of misalignment caused by poor georefering accuracy of sensors is time consuming, and some automatic solutions have been developed: these techniques are related to the image-to-image registration field on the one hand and to the vector-to-vector conflation domain on the other hand

There are three main difficulties to deal with. As stated before, the first is to reach the best alignment of the image on the ground. This so-called orthorectification process requires a geometric sensor model, and a terrain model (earth geoid, and a DEM - digital elevation model, such as (Shuttle Radar Topography Mission, n.d.)). Secondly, the main difficulty to cope with is to match features on the map with pixels from the image. Depending on the resolution of the image, the different features (road, building, natural area, etc.) cannot always be matched. Some details may be hidden, or sometimes the contrast does not allow to discriminate the different objects. The last potential issue is that the vector dataset may be outdated: for instance, buildings may have disappear, or new roads may have been built. This last problem is close to the change detection domain.

In the image registration field, (Brown, 1992) has resumed the different transformations as follows:

- linear transformations: translation or rotation
- non-linear transformations, like warping, shearing or complex deformation

In our work, the non-linear transformation is taken into account by the orthorectification process (or, by the projection of the vector dataset in raw sensor geometry). We will therefore focus on the translation transformation only. Rotation error are unlikely and can be assimilated to translation error because of the orders of magnitude of sensor altitude, field of view and errors in viewing angles.

1.2 Review of existing methods

In the literature, vector-to-image registration has seen different approaches being developed and can be mostly decomposed in two steps: firstly extracting features from the image, secondly matching these features with equivalent features from the map.

Designing such an algorithm therefore requires to choose relevant features than can be observed in the image, and a method to compare these features to equivalent features in the vector dataset.

Some methods have been proposed using surface features (Vohra, 1999), other using roads extraction. (Doucette et al., 2007) has focused on automated or semi-automated road extraction to register onto QuickBird imagery, while (Chen et al., 2004) has applied a grid matching algorithm to cross-roads extraction. (Avbelj et al., 2013) has proposed a method based on line detection in a DSM (digital surface model) and hyperspectral images. The objective is to fuse material discrimination (for instance, difference between a building and the pavement around it) and height, by detecting roof edges. These methods seem to be adapted to their context (resolution of the target image, type of features in the map) but need a prior-knowledge of what the image contains.

Our proposed method differs by the kind of features we extract: we try to reach a compromise between features that are really informative (like free form curve or closed shape features) but

*Corresponding author

uneasy to match or extract in the vector dataset and simpler, less informative features (points, small segments). Our approach is based on line-matching and segments filtering. Line-matching had proven to be suitable for 3D reconstruction (Baillard et al., 1999) with multiple views.

The remainder of the paper is organized as follows. Section 2 will detail the method. Section 3 will present the implementations details and section 4 evaluate performances on different scenes. Finally, we will discuss remaining issues and future work.

2. METHOD DESCRIPTION

2.1 Overview

Our method targets images from sub-metric resolution to a range of 10/20m resolution, and datasets that could be as detailed as building layers from Open Steet Map for registration with sub-metric images, or landuse/landcover layers for lower resolution images.

It takes advantage of an efficient Line Segment Detection algorithm (Gioi et al., 2010) to extract a lot of segments from the image, and then filter them to find corners (for instance, roof top, cross roads, footprints of buildings) to match with corners of the vector dataset.

Corners are a good option to achieve vector-to-image registration: they are present in most human constructions (buildings, cross-roads) and in most agricultural landscapes. They can be combined to form a lot of geometric shapes (if we apply a certain tolerance on the angle) and can be quite easily detected, by correctly filtering segments.

2.2 Input data

The inputs of our method are a target image (panchromatic or multispectral) and some vector dataset we want to co-register. The dataset can contain lines or polygons, but as the method is based on corners detection, the database must contain some perpendicular segments (with a certain tolerance on the angle). Thus, it is recommended to work with layers containing polygons.

2.3 Prerequisites

Due to the principles of satellite image acquisition, we must deal with different sources of misalignment. The main component of this misalignment, coming from slowly evolving angular errors in sensor modeling, can be approximated by a general offset between the dataset and the image, but we can also observe non-linear deformation (kind of warping), when acquiring an image with a high incidence angle, or when working on hilly landscapes.

The first step of our method is to get both data in a common geometry: we can either work in ground geometry or in raw sensor geometry.

The most common use case is to work in the carthographic projection of the vector data: we therefore need some terrain information (earth geoid and a digital elevation model) to perform orthorectification of the image in this projection.

Note that our method also supports transforming georeferenced vector database in the sensor geometry.

Either way, sensor model and terrain inaccuracies lead to misalignment as it can be observed in Fig. 1. However, as we will see in the evaluation in section 4, even in poor conditions leading to high misalignment, the method succeeds in registering the dataset onto the target image.

2.4 Line segment detection

The second step of our method is to perform Line Segment Detection (LSD) in the target image. This step is an intensity-based a-contrario algorithm for the detection of segments in the image (Gioi et al., 2010). Line Segments Detection aims at detecting linear details, or local zones where the gradient aligns significantly with respect to a background noise model.

Of course, the linear details detected on an image depend on the resolution, and the number and relevance of the detected segments can vary from one scene to another (as shown in Fig. 2 and Fig. 3).

In next steps, the method only processes vector datasets: the raw line segment detection and the initial database to register (polygon or line features).

2.5 Segments filtering

The aim of this step is to filter and select relevant segments in order to identify corners such as building corners, crossroads, etc. Because the complexity of the registration step is $O(n^2)$, n being the number of segments to match, it is important to find the smallest possible dataset that contains the most relevant segments, so as to reduce processing time. Those filters will be applied to extracted segments from the image (Fig. 4), as well as to segments from the database. If the vector database contains polygons, we first decompose them in individual segments.

We have implemented several filters applied sequentially on the set of segments. First filter is to apply a length criterion, to eliminate segments that are too small or too long. The complexity of this step is linear.

Then, we select segments that lies in a close neighborhood of the vector dataset. Segments too far from the database to register will not be relevant to estimate a transformation. To that extent, a distance threshold is used: it can be seen as an upper bound of the initial mis-registration. This step is $O(n^2)$, but can be optimized by using spatial indexing techniques.

The third filter is to select perpendicular segments, by forming pairs of segments that are close enough and form a right angle. This step requires two thresholds: threshold on distance

Figure 1. Projection of an OSM building layer on a PHR image of Bombay

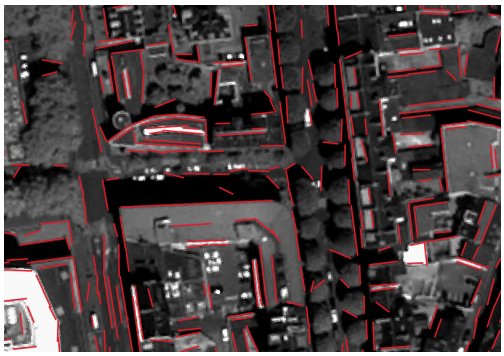

Figure 2. Detected segments on an extract of a PHR image (0.5 m resolution)

Figure 3. Detected segments on an extract of Sentinel2 image (10m resolution)

between segments, and tolerance threshold on the angle. The latter is important when working on images in raw geometry, where right angles can be tilted by viewing angles effect, and also to deal with corners with different angles (for example, to keep segments of a hexagonal building). As Fig. 5 shows, around 5% of the segments are selected for the next algorithm steps.

Figure 4. Line segment detected on a PHR image of Bombay

2.6 Translation estimation, with tuples matching

At this step, we have obtained four sets of segments, two of which being used to compute the final transformation:

- the initial database, that aims to be registered (translation)
- the raw line segment detection (to compute precision and match rate, see below)
- filtered segments from the database (to be matched to their counterparts coming from the line segment detection)
- filtered segments from the line segment detection

In order to estimate the best translation, we start by creating tuples of close segments and their intersection. This has to be done for the vector filtered from the database and for the vectors filtered from the line segment detection (Fig. 6).

The tuples issued of the database are compared to those from the line segment detection: if two tuples are close enough, a candidate transform is defined as the translation between the tuples corners. Then we compute its score, which is defined by the sum of distances between matched segments (inliers).

The best transform minimizes that score and is applied to the database.

2.7 Evaluation metrics

In a general case, we do not have a baseline (for instance, geodetic reference points) to evaluate performance of the method. So we use some unsupervised metrics to evaluate the performance of the algorithm. Those metrics are computed thanks to the registered database and the segments detected in the image. Two metrics are computed for each polygonal shape of the database:

- The match rate is a ratio of total length of matched segments with the perimeter of the feature. This can be interpreted as a reliability score (Fig. 7).
- The precision score is the mean distance between the segments of the initial feature and their counterparts in the line segment detection. This gives an accuracy of the transformation.

These metrics are also declined as global metrics: the percentage of features being matched (global match rate), and the mean precision score. Additionally, we also compute the rate of matched features, and the mean precision score. As stated before, we apply the same linear transformation to all the features of the database. So in certain cases, some features have a weak match rate but are correctly registered. It may happen for quite small features, whose segments have not been detected properly.

In other cases, a low match rate can also indicate that the building disappeared.

3. IMPLEMENTATION

Our implementation is built upon Orfeo ToolBox (Grizonnet et al., 2017), which provides many tools to handle most kinds of

Figure 5. Line segment filtered

Figure 6. The tuples from the database are compared to those filtered from the line segment detection

Figure 7. Metrics computation after registration of the vector dataset

satellite images and dataset. Furthermore, it provides a vector dataset processing framework that had been extended to implement segment filtering and segment matching applications. OTB provides a remote module mechanism that can be used to extend it and benefit from internal mechanisms of core OTB (such as streaming, projection, etc.).

Thus, our method had been implemented as a remote module, containing several applications (marked as (*) in Tab. 2) and a main Python script to chain them all in a user-friendly interface.

3.1 Projection of vector dataset in a specific geometry

Orfeo ToolBox provides all the tools to manipulate images from main sensors, in its raw sensor geometry or in a ground-based geometry (OTB Team, 2020). Thanks to the *VectorDataReprojection* application, one can project any vector dataset in any geometry. The OrthoRectification application can perform orthorectification of a satellite image to transform it into a ground-based geometry.

3.2 Line segment detection

Orfeo ToolBox contains an implementation of the Line Segment Detection algorithm, and the authors also published an open-source implementation (Gioi et al., 2010), which we embedded in a remote module. As shown in Tab. 1, this implementation detects more segments in a lower computation time than OTB own implementation. We therefore selected this implementation to evaluate our method.

3.3 New applications developed for registration

The Tab. 2 lists all the OTB applications that implement our method. All of these applications can be tuned with many parameters (the main ones are listed). For example, the *SegmentsFiltering* application can take into account different thresholds

Implementation used	Number of segments detected	Computation time (s)
OTB LineSegmentDetection	5504	42
Line Segment Detection Algorithm (Gioi et al., 2010)	13443	4

Table 1. Computation time for a 2200x1800 pixels extract of a PHR image

on the various distances that are computed: it is very useful to tune the whole method.

In the main Python script, we have fixed default values for all parameters so it only needs a target image and a vector dataset to register. But if one knows that the initial offset is larger (for instance, more than 20m in a VHR image with 0.5m resolution), it is possible to increase the *a priori* maximum distance between the projected database and the image.

4. EXPERIMENTS

4.1 Overview of the evaluation method

The evaluation method consists in projecting an image in a cartographic projection, evaluating the initial offset with the dataset, and then use our registration method to correct misalignment. For each use case, we compute unsupervised metrics introduced in 2.7, such as a global match rate (% of features being matched), a match rate per feature, a mean precision score (evaluated remaining offset).

4.2 Datasets used

As stated before, our method could be applied to different kinds of satellite images: from very high resolution (VHR: 0,5 meter or below) to middle class resolution (Spot 6) or even low resolution (Sentinel 2).

In this work, we focus on VHR images (Tab. 3) like Pleiades High Resolution (PHR) images or World View (WV) images, because it is easier to find vector dataset that have the same level of details and show some misalignment to correct.

The acquisition geometry also differs between the different images.

We have registered building layers from Open Street Map (OpenStreetMap, n.d.). The initial offset ranges from a few meters to 150 meters in an extreme case. It should be noted that the Open Street Map building layer is not always accurate: sometimes the building shape matches exactly its footprint whereas in other cases, it matches the building block, including for example a surrounding place or garden.

For the IARPA use case, we extracted buildings footprints from the LIDAR image (Fig. 8).

4.3 Results

As shown in Tab. 4, the global match rate is very high for urban scenes (Buenos Aires, Bombay, Rennes), because all buildings are correctly detected with the line segment detection algorithm: the match rate for each feature is always higher than 70% and the precision score is between one and three pixels.

OTB Application	Functionality	Main parameters
VectorDataReprojection	projects vector dataset in any geometry	
LineSegmentDetection	detects linear features	
SegmentsFiltering (*)	filters segments with various criteria	Distance and angle threshold, distance to an existing database
ImageToDBRegistration (*)	computes vector to vector conflation to estimate the best transform	Distance between tuples
VectorDataEvaluation (*)	computes metrics	
***ImageToDBRegistration* Python script (*)**	chains the different steps above	Maximum *a priori* distance to an existing database, Distance between tuples

Table 2. Orfeo ToolBox applications. (*) have been developed for this method

Use case	Satellite images (resolution)	Database
Buenos Aires (IARPA Challenge)	44 WV panchromatic images (0,3 m) + 3 PHR panchromatic images (0,5 m)	Buildings footprint extracted from a LIDAR image
Bombay	2 PHR pansharpened images (0,65m), 6° and 20° incidence angle	Buildings layer from Open Street Map
Rennes	PHR panchromatic image (0,5 m), 2° incidence angle	Buildings layer from Open Street Map
Haiti	2 PHR pansharpened images (0,5 m), 3° incidence angle	Building layer

Table 3. Description of the datasets

4.3.1 Robustness to the line segment detection In the Haiti use case, the global match rate is low (16-18%). The images have been acquired after Sandy hurricane and most buildings of the area have been affected. Furthermore, most of them are small and very close to vegetated areas. These different reasons explain the poor detection of the buildings compared to other cases (Fig. 9).

Nevertheless, the registration gives satisfying results, which can be explained by the fact that we only need one correct match to estimate the correct translation, and thus some of the larger buildings present in the dataset allows to estimate the transformation correctly (Fig. 10).

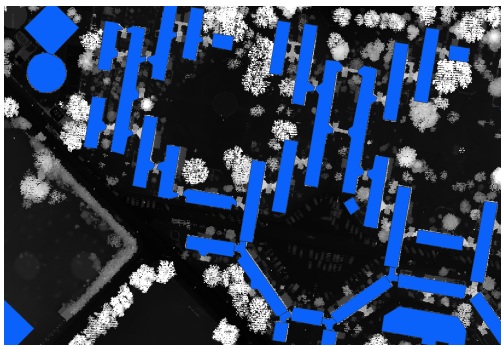

Figure 8. Ground-truth extracted from LIDAR image

Use case	Initial offset (m)	Match rate (% features matched)	Mean match rate (/feature)	Precision score (m)
Buenos Aires (IARPA Challenge)	0 to 7.5 m (mean: 3.8 m)	100 %	71% to 98%	0,46 to 0,93m (mean: O,61m)
Bombay	22.5 to 150m	100%	84%	1.5m
Rennes	8m	99%	82%	1.5m
Haiti	3m	16 & 18%	30%	0.49m

Table 4. Results of the registration method on VHR images

Figure 9. Line segments detection in an extract of Haiti case

4.3.2 Performance Computation time directly depends on the number of tuples to compare in the two different sets. On the urban scenes here mentionned (Buenos Aires, Rennes, Bombay), the input image is about 3000x3000 pixels. The line segment detection produces more than 10000 segments. If the input database has around 100 buildings, the algorithm tries to match a few hundreds of tuples from the database, with a few thousands filtered from the line segment detection. In that case, it takes less than 2 minutes for the whole process of registration.

4.3.3 Influence of the acquisition angle A lot of VHR images are acquired with a significant roll angle (up to 35°). This causes a shearing in the perspective, and in urban areas, it leads to a lot of occlusions. The robustness of the algorithm with respect to the roll angle has been investigated with the IARPA dataset, which contains a wide range of angular conditions: even on very tilted images, the method can match and precisely register the buildings, as shown in Fig. 11. The mean match score is between 0.9 and 1, whereas the mean distance is between 0.5 and 0.9 meter.

Figure 10. Registration of the database in Haiti: buildings outlined in red did not match any segment

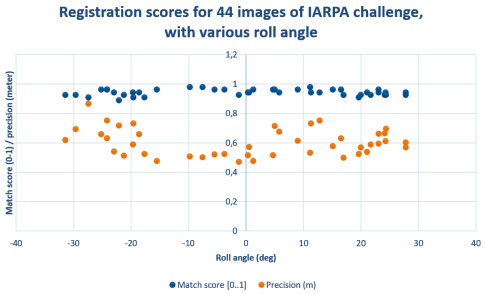

Figure 11. The match rate and the precision score do not depend of the incidence angle

5. DISCUSSION AND FUTURE WORKS

Our method has been successfully tested on urban areas, for example to register Open Street Map building layers on some Very High Resolution images, such as Pleiades, or Worldview. We have tested the method on various scenes, with initial conditions ranging from good (e.g. 5 meters offset) to very poor (e.g. 150 meters) and the algorithm always succeeded in registering vector data and image together. The *a priori* maximum distance to the projected database parameter allows to succeed in the worst case.

The method gives promising results and the algorithm only needs a weak prior knowledge of the features to register. The main parameter to tune is the proximity threshold between segments, when filtering pairs of segments to form tuples. As a default value, this parameter is set to ten times the spacing of the image. In terms of computation time, the segments filtering step is of first importance, and a local spatial filtering of the segments would help to improve computation time on large scenes.

However, we have observed that on urban areas, the line detection algorithm often detects buildings by their roof borders and in some cases (e.g. tilted images of a scene with buildings of different heights), some features of the database may need another correction: a local registration algorithm can help to improve the final registration. This local registration algorithm is not yet fully automated.

Finally, our method could be used for the purpose of changes detection: if a feature of the database has a low match rate, it may be seen as an evidence that this feature has changed. It could therefore be used for map updates.

In this work, we mainly focused on VHR images and precise buildings layers. We consider it can also be used to register landuse databases on lower resolution images (such as Sentinel-2 - 10/20m resolution), given that cultivated fields have corners that could be matched with detected segments in the image.

All the tools, and some example data will soon be contributed as open-source external remote module for Orfeo-ToolBox.

REFERENCES

Avbelj, J., Iwaszczuk, D., Müller, R., Reinartz, P., Stilla, U., 2013. Line-Based Registration of DSM and Hyperspectral Images. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W1, 13-18.

Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A., 1999. Automatic Line Matching And 3D Reconstruction Of Buildings From Multiple Views. *International Archives of Photogrammetry and Remote Sensing*, 32.

Brown, L., 1992. A survey of image registration techniques. *ACM Computing Survey*.

Chen, C.-c., Knoblock, C., Shahabi, C., Chiang, Y.-Y., Thakkar, S., 2004. Automatically and accurately conflating orthoimagery and street maps. *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, 47- 56.

Doucette, P., Kovalerchuk, B., Brigantic, R., Seedahmed, G., Graff, B., 2007. A Methodology for Automated Vector-to-Image Registration. 9-14.

Gioi, R., Jakubowicz, J., Morel, J.-M., Randall, G., 2010. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE transactions on pattern analysis and machine intelligence*, 32, 722-32.

Grizonnet, M., Michel, J., Poughon, V., Inglada, J., Savinaud, M., Cresson, R., 2017. Orfeo ToolBox: Open source processing of remote sensing images. *Open Geospatial Data, Software and Standards*, 2(1), 15.

OpenStreetMap, n.d. https://www.openstreetmap.org/.

OSM, L., 2015. Correcting imagery offset. https://learnosm.org/en/josm/correcting-imagery-offset/.

OTB Team, 2020. Orfeo toolbox documentation. https://www.orfeo-toolbox.org/CookBook/index.html.

Shuttle Radar Topography Mission, n.d. https://www2.jpl.nasa.gov/srtm/.

Vohra, V., 1999. Map-image registration using automatic extraction of features from high resolution satellite images. PhD thesis.