Novel Approaches for Aligning Geospatial Vector Maps

Mohamed Abderrazak Cherif [1], Sebastien Tripodi [2], Yuliya Tarabalka [3], Isabelle Manighetti [4], Lionel Laurore [5]

[1] Université Côte d’Azur, Géoazur – Sophia-Antipolis France - mohamed.cherif@geoazur.unice.fr
[2] LuxCarta - France - stripodi@luxcarta.com
[3] LuxCarta - France - ytarabalka@luxcarta.com
[4] Observatoire de la Côte d’Azur, Géoazur – Sophia-Antipolis France - manighetti@geoazur.unice.fr
[5] LuxCarta - France - lionel@luxcarta.com

Commission II/WG 4

KEY WORDS: Alignment, Geospatial maps, Polygonal features, optimization, self-supervision, epipolar-geometry

ABSTRACT:

The surge in data across diverse fields presents an essential need for advanced techniques to merge and interpret this information. With a special emphasis on compiling geospatial data, this integration is crucial for unlocking new insights from geographic data, enhancing our ability to map and analyze trends that span across different locations and environments with more authenticity and reliability. Existing techniques have made progress in addressing data fusion; however, challenges persist in fusing and harmonizing data from different sources, scales, and modalities. This research presents a comprehensive investigation into the challenges and solutions in vector map alignment, focusing on developing methods that enhance the precision and usability of geospatial data. We explored and developed three distinct methodologies for polygonal vector map alignment: ProximityAlign, which excels in precision within urban layouts but faces computational challenges; the Optical Flow Deep Learning-Based Alignment, noted for its efficiency and adaptability; and the Epipolar Geometry-Based Alignment, effective in data-rich contexts but sensitive to data quality. In practice, the proposed approaches serve as tools to benefit from as much as possible from existing datasets while respecting a spatial reference source. It also serves as a paramount step for the data fusion task to reduce its complexity.

1. INTRODUCTION

Over time, technological advancements, from aerial photography to satellite imaging, have revolutionized the way we capture and represent geographical data. This progression not only enhanced the accuracy but also increased the scale and frequency of data collection, making it a vital tool for various applications today. This rapid progress has given us access to a wealth of geospatial data, varying in scale, resolution, and modality. Despite their inherent value, the utility of maps is contingent upon their accuracy, detail, and relevance. While they enable the development of detailed and comprehensive maps, they also require sophisticated methods to combine data from different sources accurately. However, misalignments between different types of maps can hinder accurate interpretation and analysis. The main research question this study addresses is: How can we align different maps where misalignment can be due to various causes, such as differences in perspective, variations in map interpretation (e.g., manual vs. automatic digitization), or changes in maps over time?

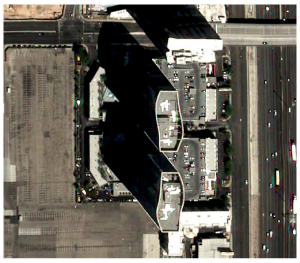
The significance of addressing this problem lies in the fact that accurate map alignment is crucial for geospatial map fusion. Aligning maps effectively reduces the complexity of matching between maps, which is an essential step in the fusion procedure. Additionally, map alignment can serve as a useful tool for map creators who want to utilize legacy maps by aligning them with updated data, allowing them to build upon existing information rather than starting from scratch.

This study delves into a wide range of methods and concepts related to map alignment, focusing primarily on vector maps representing specific features with polygonal geometries such as building maps.

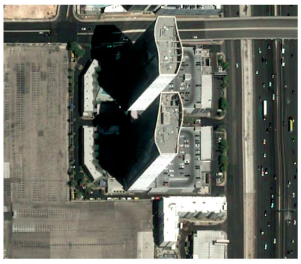
2. CAUSES OF MISALIGNMENT

Misalignment in geospatial maps is a prevalent issue, primarily attributable to various causes spanning different domains and scales. A comprehensive understanding of these causes is vital for developing effective and reliable alignment techniques. Following is a set of the major causes of misalignment that we are aiming to tackle.

- Differences in Perspective: The perspective from which geographical information is acquired plays a crucial role in shaping the final product. For instance, a map sourced from an aerial survey would have a fundamentally different perspective compared to one derived from satellite imagery. Also, the angle at which geospatial data is acquired can significantly impact the geometric properties of the map generated from that data as illustrated in Figure 1. This effect, commonly referred to as ”relief displacement”, can cause the same geographical feature to be represented in different locations on different maps, thereby leading to misalignment.
- Variation in Map Interpretation: The subjective interpretation of geographic features can result in significant discrepancies between different maps, particularly those digitized manually. This subjectivity can manifest in several ways, such as variations in defining the significance of geographic features or disagreements about feature boundaries. Consequently, two maps of the same area, interpreted by different cartographers, may exhibit marked differences, thereby leading to misalignment. As such, maps generated through different automated extraction techniques or even the same technique under different



(a) Orthorectified satellite image 1 with acquisition angle a) satellite azimuth: 320 °and b) satellite elevation 65 °



(b) Orthorectified satellite image 2 with acquisition angle a) satellite azimuth: 222 °and b) satellite elevation 62 °

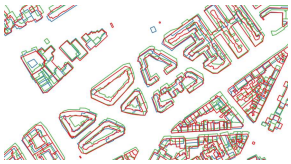
Figure 1. Illustration of perspective variations of maps with different acquisition angles, an example of high buildings over Los Angeles (USA).

parameters can have significant variations. This can include differences in the level of detail, the interpretation of complex features, or the handling of ambiguous elements within the geographical space. These inconsistencies can be exacerbated when considering the challenge of detecting changes over time. Figure 2 displays this phenomenon between a manually digitized map (subfigure c) and automatically extracted maps (subfigure a,b).

(a) Building delineation automatically extracted from satellite image

(b) Building delineation automatically extracted from satellite image

(c) Building delineation manually digitized (OSM)



(d) Overlay of vector maps

Figure 2. Visualization of different maps building delineation over Paris. (a) and (b) shows the automatically extracted maps from satellite images (c) shows the manually digitized vector map over Google satellite image as background (basemap), (d) is an overlay of all vector maps where one clearly can perceive the positional and geometrical dissimilarity.

- Temporal Variations: Geographical landscapes are dynamic and evolve over time due to both natural and human-induced changes. Consequently, maps created at different times can have significant disparities. Rivers change their courses, roads, and structures are built or demolished, and land use patterns alter over time. These temporal variations can result in misalignment when comparing or merging maps from different periods. Figure 3 depicts a visually noticeable temporal variation due to human-induced change.

- Data Processing Variations: Different methodologies and algorithms used in the processing and preparing maps for final use can induce discrepancies. For example, variations in the orthorectification process, which corrects satellite imagery for tilt and terrain displacement, can result in misalignment between maps.

(a) Satellite image at construction time of the buildings

(b) Satellite image after construction of the buildings

Figure 3. Illustration of temporal variations of maps; an example of a reconstructed building from satellite images at different timestamps.

3. PROBLEM DEFINITION

Given a vector map of polygons, henceforth referred to as the ”to-align” map, and a reference map of building features deemed spatially more accurate, the goal is to determine an appropriate transformation function to apply to the group of polygons in order to increase spatial similarity. The reference map can either be a vector map or a probability map derived from an automatic feature extraction model. Our objective is to find an affine transformation per geometry or more precisely a translation transformation since we assume that input maps have the same scale and same orientation, preserving the structural integrity of the geometries. The optimal transformation is measured by the fitness of the transformed geometry within the reference map.

4. RELATED STUDIES

4.1 Traditional Methods for Geospatial Map Alignment

Traditional approaches to geospatial map alignment primarily revolve around manual techniques and rule-based algorithms. These methods often utilize geometric transformations and control points to align features in different maps.

- Geometric Transformations: This method involves using affine or projective transformations to align features based on manually selected control points. While effective for small-scale adjustments, this approach can be labor-intensive and prone to human error.

- Rubber Sheeting: A technique that stretches or compresses map features to align with a reference map. This method can be effective for local adjustments but may not maintain the integrity of geometrical properties over large areas as shown in (Doytsher, 2000) and (Sun et al., 2020).

- Feature-Based Alignment: Involves aligning maps based on shared features, such as road intersections or landmark buildings. This method is dependent on the availability and accuracy of shared features, limiting its applicability as studied in (Zhang et al., 2018).

4.2 Learning-Based Methods for Geospatial Map Alignment

Recent advancements in machine learning, particularly deep learning, have introduced new methodologies for map alignment. These methods leverage large datasets to learn complex patterns and transformations, enabling more accurate and efficient alignments.

Deep Learning for Cadastre Map Alignment (Girard et al., 2018) proposed a multi-task, multi-resolution deep learning framework that aligns existing building polygons to new images and detects new buildings. This method outperforms traditional alignment methods, especially in handling large and complex datasets. The approach uses neural networks to predict displacement fields, offering significant improvements in alignment accuracy and the ability to update maps with new constructions.

MapRepair for Temporal Inconsistencies (Zorzi et al., 2020) introduced an end-to-end deep learning approach, MapRepair, to align and correct temporal inconsistencies in cadastre maps. This method uses a neural network to generate aligned cadastre masks and segment new buildings, effectively managing misalignments and updating maps to reflect recent constructions. It demonstrates superior performance in dealing with heavily distorted annotations, a common issue in traditional methods.

4.3 Comparison and Integration

Comparing traditional and learning-based methods reveals distinct advantages and challenges. For instance, learning-based methods generally offer higher accuracy, especially in complex and large-scale scenarios, due to their ability to learn from extensive datasets. While traditional methods can be time-consuming and require manual intervention, learning-based approaches automate the alignment process, significantly reducing the time and effort required. Also deep learning models, once trained, can adapt to a variety of scenarios and datasets, whereas traditional methods may need manual adjustments for different contexts. Learning-based methods require large, annotated datasets for training, which can be a limitation in areas where such data is unavailable or of poor quality. Finally, traditional methods, being rule-based, often offer more interpretability compared to the ’black-box’ nature of deep learning models, which can be a concern in certain applications.

In conclusion, while traditional methods retain relevance for specific, small-scale applications or in scenarios with limited data, learning-based approaches, as exemplified by (Girard et al., 2018) and (Zorzi et al., 2020), represent the forefront of technology in map alignment, offering scalability, accuracy, and efficiency. The integration of these methods can lead to more robust and versatile map alignment solutions.

5. PROXIMITYALIGN: ENERGY OPTIMIZATION BASED ALIGNMENT

5.1 Approach overview

Figure 4 shows an overview of the proposed pipeline which is composed of a preprocessing step, a clustering step, and an energy minimization step.

The first step in our proposed pipeline is to preprocess the reference map into a georeferenced raster map of the polygon contour’s proximity which represents the distance of each pixel in the map to the nearest contour of a polygon. For vector input maps, we apply a process of rasterization on polygon contours, followed by the generation of a proximity raster from the binary rasterized map. For probability maps, we apply a thresholding operation on the contours’ probability band.

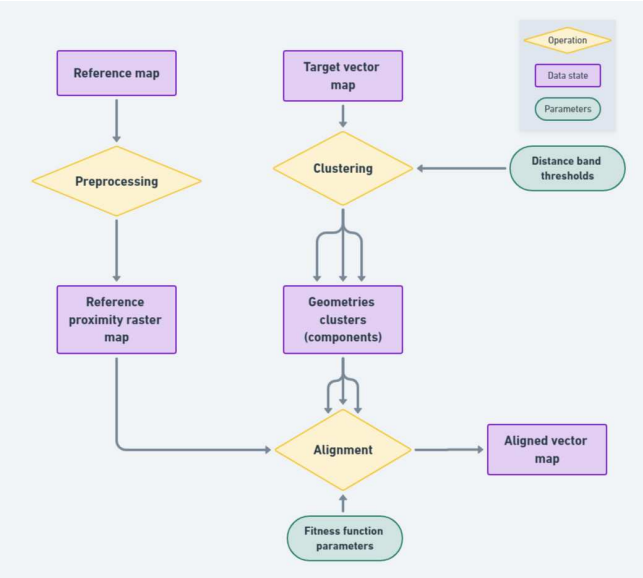


Figure 4. Overview of the ProximityAlign pipeline.

Given a user-defined list of decreasing distance values, we proceed to cluster geometries into groups of geometries specified as ”components”. Each distance value corresponds to a stage in the alignment algorithm generating different sized components at each level. Figure 5 shows an example of clustering results at different stages.

In each stage, we try to find the optimal translation vector for each component, which is a two-dimensional derivative-free optimization search problem. We do this by measuring the fitness of transformation per geometry. Following this, the geometries are positioned for the next search with a new set of components. This approach allows us to mimic the pyramidal or coarse-to-fine process of alignment, ensuring both accuracy and computational efficiency.

5.2 Geometry fitness function

The alignment process is focused on minimizing the distance between the geometries of the input and reference maps. With the aid of the proximity map, an ideal alignment is considered as a translation that moves a geometry so that the sum of proximity map pixels, which fall under its contour, is zero. A basic fitness function for a geometry would be the sum or mean of pixel values underneath the polygon contour ring, let’s consider $PV(dX, dY)$ as the ensemble of proximity map pixel values corresponding to the respective set of segments after translation using the vector with (dX, dY) components. We propose the following weighted energy function which is composed of three components, each with a different purpose:

- Mean-based Component: This component of the function measures the average distance to the reference geometries of pixel values underneath the contour. The sigmoid function allows us to place higher importance on lower mean values, as they indicate a better alignment.

(a) Connected components using distance band with a maximum distance equal to 5 meters

(b) Connected components using distance band with a maximum distance equal to 2 meters

Figure 5. Illustration of building geometrical components using different thresholds. Connectivity edges between geometries in the same component have the same color. Due to the large distance threshold, large components appear in subfigure (a) like the yellow or green components. After decreasing the distance threshold as shown in subfigure (b), components are smaller and have a limited span.

- Partial continuity Component: This component computes the ratio of pixel values that fall under a certain threshold. It counts the lengths of contiguous subsequences of pixels with values less than the threshold. A lower ratio is preferred as it indicates a better partial alignment.
- Variability Component: This component measures the homogeneity of contour alignment by measuring the standard deviation on the ensemble of pixel values, as some translation candidates can result in partially aligned polygons.

The three components are combined with weights α, β, and γ (where $\alpha + \beta + \gamma = 1$) to give the final fitness function.

$$E_{fitt} = \frac{\alpha \times E_m + \beta \times E_c + \gamma \times E_v}{\alpha + \beta + \gamma} \quad (1)$$

6. OPTFLOWALIGNMENT: VECTOR MAP ALIGNMENT VIA OPTICAL FLOW

Optical flow estimation methods are based on the assumption that the pixel intensities of an object do not change between consecutive frames, which is rarely the case for optical satellite images due to illumination and shadow effects, therefore we opt to estimate optical flow between segmentation probability maps since they do not display intensity variation, as these maps are generated using a robust deep learning model that is less prone to illumination effect, thus respecting the major assumption of intensity uniformity between images. Figure 6 shows an example of intensity variation between optical images for the same feature. We note that when considering segmentation maps as reference images to estimate optical flow between different perspectives we gain the uniformity of intensities but we lose the texture peculiarity of real-world features.

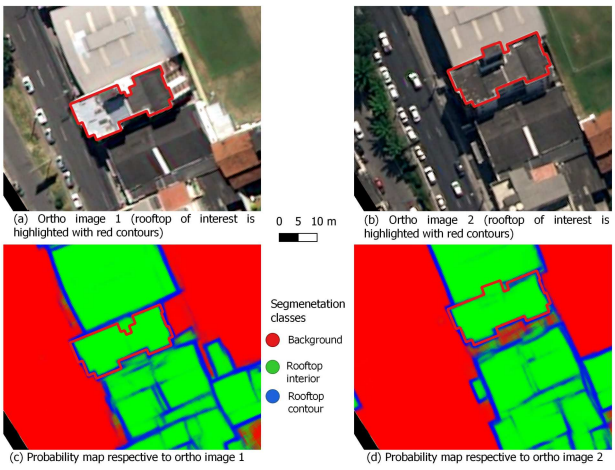


Figure 6. Example of intensity variation between a couple of optical images (top) cropped over a building rooftop, each optical image has its’ respective segmentation map (bottom).

We propose a deep-learning approach to estimate and generate a displacement map between two segmentation images of buildings’ rooftops (segmentation maps are composed of three classes 1) rooftop interior, 2) rooftop contours, and 3) background). We employ the deep flow model FlowNet described in (Fischer et al., 2015) and PWC-Net (Sun et al., 2018) and train them in a self-supervised manner, as given vector maps of features (buildings rooftop in our case) we can form a large dataset composed of pairs of reference and fake misaligned maps. In their study (Shah and Xuezhi, 2021), the authors pointed out that varying illumination, large displacement, and lack of texture are significant challenges in the optical flow field. While the first challenge is handled by using segmentation maps instead of optical images as inputs, the rest of the challenges are approached by the choice of the model architecture, as both architectures are spatial pyramid networks that estimate large motions in a coarse-to-fine manner by propagating estimated flow at different levels of scale. The models generate a displacement field map as shown in Figure 7 that will be used to transform misaligned geometries in a rigid or non-rigid way.

6.1 Mathematical modeling

Given two images of segmentation probabilities (indicating for each pixel whether it belongs to a specific feature class) of the same size H x W, with R as the reference image and S as the subject image of misaligned features. Inputs could also be vector maps as they could be transformed into probability maps through rasterization. The alignment problem aims at finding a deformation, i.e. a 2D vector field $\mathbf{g}$ defined on the discrete

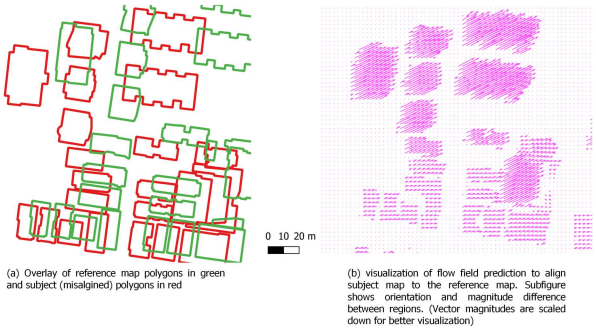


Figure 7. Sample flow field prediction using FlowNet model.

image domain [1, H] × [1, W], such that the warped second image $S \circ (Id + g)$ is well registered with the first image R or in other words, the warped geometries extracted from the subject probability map S are superimposed on geometries extracted from the reference probability map R. To train a deep learning model to estimate the deformation $\hat{g}$, a ground truth g_{gt} must be provided for the supervision task.

6.2 Synthethic dataset generation

Given a vector map of polygons, we generate a new vector map by randomly displacing the geometries at different levels of clustering. Throughout this process, we track the displacement values associated with each geometry. Then we rasterize the displacement values to create ground truth maps at varying levels of resolution. We also introduce randomness in the displacement of the geometries to ensure diversity in orientation. Additionally, we create probability maps with different resolutions and vary the maximum displacement to achieve magnitude diversity in the synthetic dataset. To ensure further diversity and realism, we consider specific parameters and variations in the generation process. We incorporate clustering parameters and set maximum displacement values for each cluster level. By fine-tuning these parameters, we can control the level of displacement and achieve a wider range of realistic scenarios. It is important to note that the rasterized ground truth optical flow is always superimposed on the second (subject) image as shown in Figure 8. This ensures that the optical flow information aligns with the subject image and facilitates the training process.

6.3 Training loss

Given θ the set of all learnable parameters for each network, which for FlowNet includes (image encoder branches, correlation encoder, and decoder layers), and for PWCNet includes (feature pyramid extractor and optical flow estimators at each level). Let W_θ^l denote the flow field at the lth pyramid level predicted by the network and W_{GT}^l the corresponding ground truth supervision optical flow. We employ the multi-scale loss function proposed in (Fischer et al., 2015):

$$\mathcal{L}(\theta) = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} \left| \mathbf{w}_\theta^l(\mathbf{x}) - \mathbf{w}_{\mathrm{GT}}^l(\mathbf{x}) \right|_2 + \gamma |\theta|_2 \tag{2}$$

Where $|\cdot|_2$ is the L2 norm of a vector, and α_l is the weight of each level error.

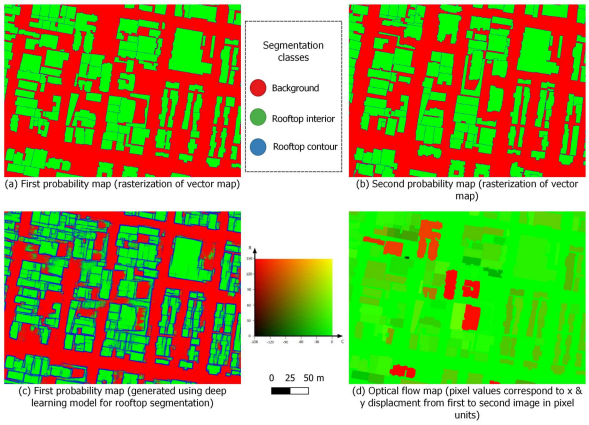


Figure 8. Example of a generated training dataset using geometry matched dataset. The first two subfigures correspond to rooftop probability maps created from vector maps through rasterization. Subfigure (c) corresponds to the probability map of the first view generated using a deep learning model. Subfigure (d) is the generated optical flow image after assigning vertical and horizontal displacement values in pixel units.

6.4 Hyperparameter Choices

The employed architectures present a multitude of configurations that affect training and inference in both quality and runtime aspects. The following are the major hyperparameter choices.

- Pyramid levels: Pyramidal deep learning approaches mimic the traditional coarse-to-fine methods (Black and Anandan, 1996) (Brox et al., 2004) (Sun et al., 2014), where images are processed at different scales. We notice that on high levels, lots of shape information is lost. In our case, we suffice with 4 pyramidal levels as no significant improvement was achieved with more levels. This hyperparameter is coupled with the ”max displacement” hyperparameter as they both lead to the overall maximum displacement.

- Maximum displacement per level: Both architectures present a strategy to compute the cost volume of associating a pixel to its correspondent in the second feature map, however, matching is limited within a specific range defined by $[-k, k]x[-k, k]$ where k is the maximum displacement. In our case, we set the maximum displacement to 15 for each level, resulting in **Maximum overall displacement** $= 2^{L-1} \times k = 120$ pixels.

- Encoder decoder depth: In contrast to our approach that utilizes semantic maps to predict optical flow, the authors of FlowNet (Fischer et al., 2015) and PWCNet (Sun et al., 2018) used RGB images from the KITTI dataset (Geiger et al., 2013) and the MPI Sintel (Butler et al., 2012) for optical flow prediction. This disparity in data sources introduces notable differences in the characteristics of the respective datasets. While RGB images contain rich texture and color information, enabling the extraction of high-level visual features, our dataset primarily focuses on the structural and spatial relationships between objects present in the scene. In this fashion, we reduce the complexity of both architectures as

no visual features are present in our dataset by decreasing the number of channels in both encoder and decoder modules, for instance, for the Flownet model, the authors proposed a channel configuration for the ConvEncoder to be $[64, 128, 256, 256, 512, 512, 512, 512, 1024, 1024]$ in our case a configuration of $[8, 16, 32, 64, 128]$ leads to fairly acceptable optical flow results. Decreasing the CNN layers' depths not only lowered training and inference time but also made the setting of high "maximum displacement" values possible expanding the search domain and matching very distant features.

7. EPIPOLAR GEOMETRY-BASED APPROACH

One key motivation for exploring an epipolar geometry-based approach for geospatial map alignment is the potential to reduce the computational complexity of the alignment task. Based on the geometric constraints provided by epipolar geometry, we can significantly narrow down the search space for feature matching, thereby improving efficiency. The proposed approach is not an alternative to previously proposed energy minimization and deep learning methods approaches, we can view it as a complement as each of these methods can be adapted to gain from its advantage.

Feature matching, disparity estimation, and template matching are pivotal techniques in computer vision, crucial for tasks like image registration, 3D reconstruction, and object localization (2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications | Wiley, n.d.). Feature matching involves identifying and aligning key points in images, such as corners or textures, using algorithms like SIFT (Lowe, 1999), SURF (Bay et al., 2006), and ORB (Rublee et al., 2011). Disparity estimation, or stereo matching, is essential for depth estimation and 3D scene understanding, involving steps like rectification, correspondence matching, and disparity computation (Scharstein and Szeliski, 2002). Challenges in this area include handling occlusions and texture-less regions. Template matching, on the other hand, focuses on detecting and localizing specific objects or patterns within images by comparing a predefined template across various image regions. These methods collectively facilitate accurate map alignments and scene matching in computer vision, setting the stage for further analysis and rectification using epipolar geometry.

7.1 Mathematical modeling of epipolar rectification

When working with an overlapping pair of ortho stereo images, we generally consider that these images represent a planar view of the scene. In this context, a rotational transformation applied to both images ensures that features move horizontally in the image space. The rotation angle is computed using both satellite's positions in a polar reference system using the angles of acquisitions as follows:

$$Ra = atan2(\frac{cos(Az_1)}{tan(El_1)} - \frac{cos(Az_2)}{tan(El_2)}, \frac{sin(Az_1)}{tan(El_1)} - \frac{sin(Az_2)}{tan(El_2)})$$
$$\text{Given } Az_i \text{is the azimuth angle for image } i \text{ ,and}$$
$$El_i \text{ is the elevation angle for image } i. \quad (3)$$

7.2 Alignment with Epipolar Geometry Awareness

Incorporating the acquisition angles from the reference satellite imagery used in map alignment, we can define the vector that represents the direction of displacement from the target to the reference map. This is formalized as:

$$\vec{DISP} = \begin{bmatrix} disp_x \\ disp_y \end{bmatrix} = \begin{bmatrix} \frac{sin(Az_1)}{tan(El_1)} - \frac{sin(Az_2)}{tan(El_2)} \\ \frac{cos(Az_1)}{tan(El_1)} - \frac{cos(Az_2)}{tan(El_2)} \end{bmatrix} \quad (4)$$

As indicated previously, the derivation of this vector serves to simplify the alignment challenge, transforming it from a two-dimensional problem to a more manageable one-dimensional constraint. With this adjustment, both the "ProximityAlign" method (discussed in Section 5) and the "FlowAlign" method (detailed in Section 6) can be efficiently adapted to leverage the benefits offered by epipolar geometry in case these metadata are available.

7.3 Procedural alignment through image matching techniques

We present an alignment method relying on the result of matching between optical images as guidance of displacement at the feature level in a procedural manner, where given an ordered set of techniques to estimate displacement, each geometry is aligned independently by the most prominent method, if the alignment fails, it falls back to the following method.

7.3.1 Stereo matching using semi-global matching From the input stereo pair of orthoimages, we apply epipolar rectification as described in 7.1, then, we use a modified version of SGM algorithm (Tripodi et al., 2020) to enable solving certain conditions such as large displacements and textureless regions. The used method is a buildup of the original algorithm as it is a 1) Pyramidal approach: since SGM is executed at different scales ([8,4,2,1]) thus removing noise by incorporating disparities at all levels, 2) Usage of the census as a cost function being more robust to radiometric difference, 3) Runtime enhancement as the algorithm is implemented in GPU. Figure 9 shows a dense disparity map to match pixels in the stereo pair, as the majority of the required pixels (rooftop pixels) are estimated, we still notice cases of inaccuracy, incomplete estimation for each rooftop, and total absence of disparity estimation for some rooftops. These inconveniences are due to different factors such as radiometric differences, shadow effects, and occlusion of high features. Thus, the a need for an alternative method to align remaining features with erroneous or missing disparities.

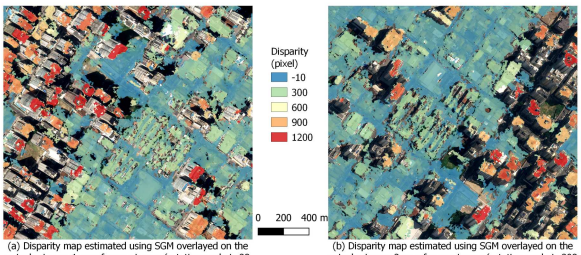


Figure 9. Example of disparity map estimation on a stereo pair (over Brazil Vila Velha) each subfigure corresponds to disparity map overlayed on its respective optical image used as reference. Maps are inversely rotated since epipolars are generated in a way that displacement is always from left to right.

7.3.2 Template matching Template matching finds the position of the most similar region to a small image patch known as template image **T** in a larger image patch known as search image **S**. We benefit from the vector map to extract the respective image patch **T** from the target optical image alongside the respective binary mask that denotes the pixels corresponding to the rooftop in the rectangular image patch. Similarly, for the search image retrieval, we use the polygon geometry extents to extract the respective search image **S** with a spatial buffer on the sides as the displacement is horizontal.

7.3.3 Procedural alignment pipeline Procedural alignment consists of successive application of disparity estimation techniques at the feature level. We define "view data" as the ensemble of data elements corresponding to a scene that is composed of the source optical image, acquisition metadata, and the corresponding vector map. The pipeline takes as input two "view data" of a stereo pair, designating the vector map of "view data 1" as misaligned regarding the vector map of "view data 2" that is considered as a reference.

An overview of the pipeline is available in appendix A. The first step consists of applying epipolar rectification of the optical images that will be used by the next alignment methods. As the first disparity estimation method, we employ the modified SGM algorithm from (Tripodi et al., 2020) to compute the pixel-wise disparity map. As shown in Figure 9, disparity values are assigned to feature pixels of the reference image, i.e., disparity values are assigned to pixels of buildings' rooftops. The "SGM disparity assignment" transformation unit, aggregates disparity values corresponding to each rooftop polygon and assigns to it two attributes: 1) disparity 90th percentile and 2) disparity completeness percentage, these attributes will contribute to the computation of alignment certainty as follows:

$$P_{certainty} = P_{disp\_comp} \times (1 - fitness(P, P_{disp})) \quad (5)$$

For every polygon feature P: P_{disp_comp} is the disparity completeness percentage, and P_{disp} is the disparity value assigned. $fitness(P, v)$ is a function to measure the fitness of the translated geometry using the disparity value v w.r.t the reference vector map, as mentioned in sec. 5.2. The codomain of the certainty function is [0,1] certainty values close to zero correspond to bad alignment while values close to 1 correspond to good alignment. Thus we define a threshold to separate features into *successfully aligned* or *non-aligned* features, in our case we define $TH = 0.5$ as a threshold. For features with low certainty scores, we move to the next disparity assignment based on template matching, similar to the previous technique, this transformation unit assigns for each polygon feature two attributes: 1) best match disparity and 2) best match correlation value. These attributes will be used to compute the current method's certainty as follows:

$$P_{certainty} = P_{disp\_corr} \times (1 - fitness(P, P_{disp})) \quad (6)$$

For every polygon feature P: P_{disp_corr} is the T patch's best correlation in the search image S. Since we use normalized Pearson coefficient correlation metric, correlation values range from 0 to 1. The minimum value of 0 indicates no match or a very weak correlation conversely, the maximum value of 1 indicates a perfect match or a strong correlation. We end up with a similar codomain of the certainty function, thus, we set $TH = 0.5$ as a threshold for the current method. In the end, the "certainty filter" transformation unit separates features into two different output layers based on the certainty score as shown in appendix A. We intend to highlight that the pipeline structure is extensible, as we can add as many blocks as intended (delimited with a dotted purple line in the pipeline diagram) to solve uncertain alignment results successively.

8. RESULTS AND EVALUATION

8.1 Dataset overview

In our efforts to thoroughly evaluate the alignment pipeline, we made use of a diverse array of testing locations. Each area chosen was unique, presenting its own set of urban characteristics and feature density factors that may significantly impact alignment results. The chosen areas varied from highly urbanized, densely populated regions to those featuring less dense, more varied architectural forms. This enabled us to examine how our alignment pipeline handled these distinct environments and allowed us to assess its robustness across varying urban landscapes. Turning to the specifics of the dataset used in the pipeline, we obtained our data from several different sources. The satellite imagery was primarily obtained from VHR imagery providers, which provided images with a resolution down to 30cm. We have curated a diverse urban dataset consisting of sub-datasets from seven distinct cities spread across different parts of the world. Each sub-dataset within the urban dataset is derived from at least two different acquisition perspectives, offering multiple vantage points of the same urban landscape. For each acquisition, a manual vectorization of building rooftops is performed, creating an accurate and detailed representation of the urban environment. In addition, an automatic extraction method described in (Bauchet et al., 2022) is applied to generate an alternative set of rooftop vectors. Table 1 encapsulates numerical statistics mirroring the extent and size of each city sub-dataset, such as the total area of each city's zone extents, the count of individual building footprints, and the overall built-up area coverage expressed as a percentage of the zone area.

Sub-datasets	Area (km^2)	Number of polygons	Built-up area (%)
Ethiopia Addis Ababa	0.8	1441	24.1
India Mumbai	4.4	8039	17.4
Brazil Vila Velha	1.1	3768	39.4
Qatar Doha	1.0	708	36.6
Pakistan Rawalpindi	1.8	1197	19.4
Kuwait Kuwait	3.2	960	11.1
Sweden Stockholm	1.5	443	13.2

Table 1. Urban buildings sub-datasets statistics

8.2 Numerical evaluation

In the case of map alignment evaluation, we can make use of "Displacement Error Measurement" where the error is quantified using Root Mean Square Error (RMSE), which serves as a direct and intuitive metric for the accuracy of the alignment. But in order to compare the current methods with previous studies that do not necessarily produce vector maps, we employ "Map Similarity Measurement" involving the rasterization of the reference and the aligned map to compute the mean Intersection over Union (IOU) of the binarized maps. By applying this similarity metric between two layers, we can derive an average, normalized similarity score. This score ranges from 0 to 1, where values close to 0 indicate high dissimilarity and a score of 1 signifies exact similarity. Table 2 in Appendix B provides

a numerical evaluation of the proposed methods and most two related studies methods.

9. CONCLUSION

Proposing three innovative methodologies devised for the alignment of vector maps, each exhibiting distinct advantages, drawbacks, and suitability across various use cases. Herein, we delve into an in-depth examination of ProximityAlign, Optical Flow Deep Learning-Based Alignment, and Epipolar Geometry-Based Alignment, shedding light on their individual merits and demerits, and potential avenues for future enhancements.

A comparative exposition reveals a spectrum of strengths and weaknesses across the methods. While ProximityAlign excels in precision albeit with a higher runtime, the Deep Learning approach offers a runtime-effective solution with easier training regimes. Conversely, the Epipolar Geometry-Based method, while efficient, is data-greedy and susceptible to orthorectification and metadata-induced errors.

The differentiated yet complementary nature of these methods hints at a future where an amalgamated framework could potentially leverage the collective strengths of these methodologies. Such an integrated framework could navigate the intricate domain of geospatial map alignment with enhanced accuracy, efficiency, and broader applicability, thereby propelling the realm of geospatial data processing and analysis forward.

ACKNOWLEDGEMENTS

This work has been supported by the French government, through the France 2030 investment plan managed by the Agence Nationale de la Recherche, as part of the Université Côte d’Azur’s Initiative of Excellence, reference ANR-15-IDEX-01.

REFERENCES

2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications | Wiley, n.d.

Bauchet, J.-P., Gobbin, A., Tarabalka, Y., 2022. From Raster Predictions to Vector Layers of Buildings: A Computational Geometry Approach. 795–798.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded Up Robust Features. A. Leonardis, H. Bischof, A. Pinz (eds), *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 404–417.

Black, M. J., Anandan, P., 1996. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. *Computer Vision and Image Understanding*, 63(1), 75–104.

Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High Accuracy Optical Flow Estimation Based on a Theory for Warping. T. Pajdla, J. Matas (eds), *Computer Vision - ECCV 2004*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 25–36.

Butler, D. J., Wulff, J., Stanley, G. B., Black, M. J., 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds), *Computer Vision – ECCV 2012*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 611–625.

Doytsher, Y., 2000. A rubber sheeting algorithm for non-rectangular maps. *Computers & Geosciences*, 26(9), 1001–1010.

Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T., 2015. FlowNet: Learning Optical Flow with Convolutional Networks. arXiv:1504.06852 [cs].

Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), 1231–1237. https://doi.org/10.1177/0278364913491297. Publisher: SAGE Publications Ltd STM.

Girard, N., Charpiat, G., Tarabalka, Y., 2018. Aligning and Updating Cadaster Maps with Aerial Images by Multi-Task, Multi-Resolution Deep Learning.

Lowe, D., 1999. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1150–1157 vol.2.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571. ISSN: 2380-7504.

Scharstein, D., Szeliski, R., 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1), 7–42. https://doi.org/10.1023/A:1014573219977.

Shah, S. T. H., Xuezhi, X., 2021. Traditional and modern strategies for optical flow: an investigation. *SN Applied Sciences*, 3(3), 289. https://doi.org/10.1007/s42452-021-04227-x.

Sun, D., Roth, S., Black, M. J., 2014. A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. *International Journal of Computer Vision*, 106(2), 115–137. https://doi.org/10.1007/s11263-013-0644-x.

Sun, D., Yang, X., Liu, M.-Y., Kautz, J., 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. arXiv:1709.02371 [cs].

Sun, K., Hu, Y., Song, J., Zhu, Y., 2020. Aligning geographic entities from historical maps for building knowledge graphs.

Tripodi, S., Duan, L., Poujade, V., Trastour, F., Bauchet, J.-P., Laurore, L., Tarabalka, Y., 2020. Operational Pipeline for Large-Scale 3D Reconstruction of Buildings From Satellite Images. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 445–448. ISSN: 2153-7003.

Zhang, Y., Yang, P., Li, C., Zhang, G., Wang, C., He, H., Hu, X., Guan, Z., 2018. A Multi-Feature Based Automatic Approach to Geospatial Record Linking. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 14(4), 73–91. https://www.igi-global.com/gateway/article/www.igi-global.com/gateway/article/210653. Publisher: IGI Global.

Zorzi, S., Bittner, K., Fraundorfer, F., 2020. Map-Repair: Deep Cadastre Maps Alignment and Temporal Inconsistencies Fix in Satellite Images.

A. PROCEDURAL ALIGNMENT PIPELINE

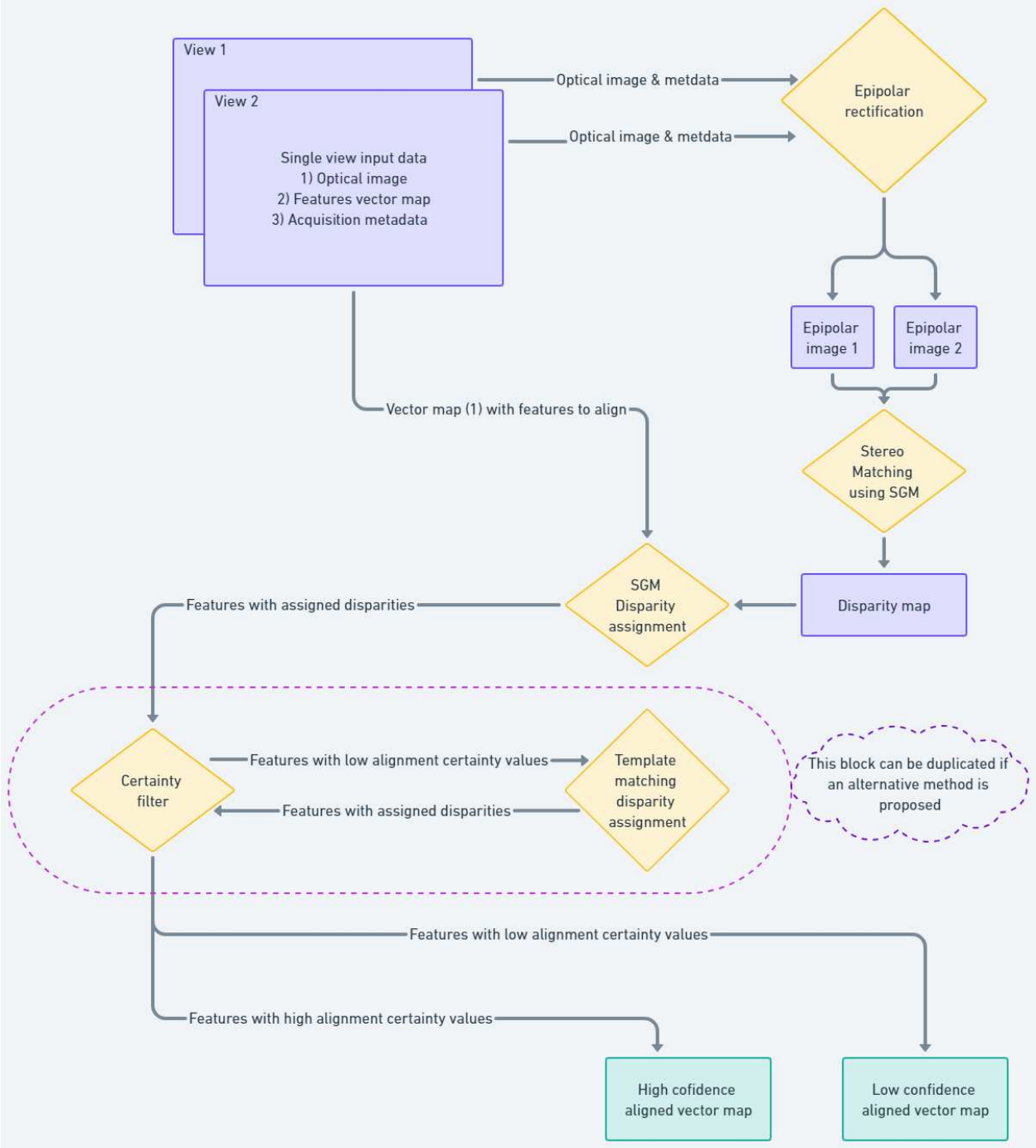


Figure 10. Overview of the procedural alignment pipeline. Blue rectangles correspond to input and intermediate transformed data, Yellow diamonds represent transformation units, and Green rectangles correspond to output data.

B. NUMERICAL EVALUATION TABLE

Datasets	Misaligned IOU	ProximityAlign method	PWCNet DL model	FlowNetC DL model	Procedural alignment	MapRepair method	MapAlignment method
Brazil Vila Velha	0.391	0.698	**0.740**	0.670	0.665	0.554	0.414
Ethiopia Addis Ababa	0.669	**0.906**	0.828	0.688	0.847	0.715	0.669
India Mumbai	0.619	**0.857**	0.846	0.687	0.796	0.702	0.662
Kuwait Kuwait	0.820	**0.957**	0.929	0.889	0.869	0.875	0.835
Pakistan Rawalpindi	0.448	**0.894**	0.860	0.713	0.676	0.690	0.449
Qatar Doha	0.594	**0.841**	0.696	0.639	0.754	0.632	0.601
Sweden Stockholm	0.564	**0.906**	0.812	0.625	0.826	0.711	0.593

Table 2. Alignment methods comparison table. Intersection-over-union (IOU) metric is used to measure the similarity between maps. The first numeric column corresponds to the IOU similarity between maps before alignment. The rest numeric columns correspond to the IOU similarity between maps after alignment using different methods.