Towards Probabilistic Georeferencing of Geological Maps from PDF Reports

Marijan Soric[1]

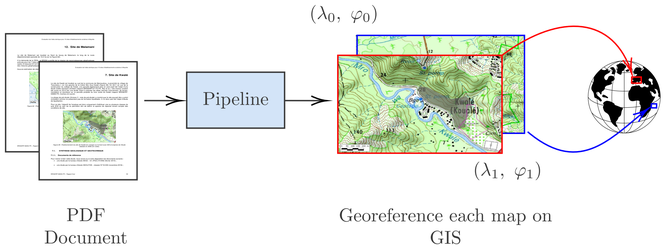
supervised by Cécile Gracianne[2] Ioana Manolescu[3] Pierre Senellart[1]

[1]DI ENS, ENS, PSL University, CNRS, Inria, Paris [2]BRGM, Orléans [3]Inria, Institut Polytechnique de Paris

Motivation

Geologists write up their fieldwork as **PDF reports**: text and **geological maps** of where they went. That map is the record of *where*, and it is locked inside the PDF.

Georeferencing puts real coordinates on a map image, so reports can be searched in a GIS. We do it *automatically*, from the PDF alone.



Prior work[1,2,3] **can only return what a database already knows**: either a map that is already georeferenced, or the gazetteer coordinates of a place name.

Problem

Let $\mathcal{I}$ be a reference map image of known physical size (W, H), with text $\mathcal{T}$ (caption, references, on-map labels). We want its **geographic centre** $x = (\lambda, \varphi) \in \mathcal{X}$, enough to place $\mathcal{I}$ in a GIS.

We **render** a candidate map $M(x)$ from OpenStreetMap[9] and score how well it agrees with $\mathcal{I}$. Georeferencing becomes a search

$$x^\star \in \arg\max_{x \in \mathcal{X}} f(x), \qquad f(x) = \text{visual agreement}\big(\mathcal{I}, M(x)\big),$$

over a *country-sized* area. f is **expensive**: every call means one render and one run of a deep matcher.

Contributions

- We build reference maps from **OpenStreetMap** as we go, instead of matching against a fixed set of maps, so we cover the whole world.
- The matcher gives us a **vector**, not just a score: how far the matched points moved becomes a *pseudo-gradient* that jumps toward the answer.
- We search with two models and compare against GPT-5.2. Pseudo-gradient descent does the work and cuts the median error; πBO stays in the loop to rescue the runs that get stuck.

Paper Personal Website

marijan.soric@inria.fr

KDD '26, Jeju, Republic of Korea

Preprocessing → spatial prior

Docling[4] reads each PDF and keeps every figure's *caption* and the sentences that mention it ($\mathcal{T}$). Candidate **place names** k come from two sources: **NER** on the text around the figure, and **mapKurator Spotter**[5], an OCR system built for maps. Both are **geocoded** (λ_k, φ_k) with **Nominatim**, after a fuzzy match against the gazetteer. A model learns weights for each candidate from OCR confidence, NER score, Nominatim importance and string distance.

$$\pi(x) = \sum_k w_k \mathcal{N}_2\big(x \,\big|\, \mu_k, \Sigma_k\big), \qquad \mu_k = (\lambda_k, \varphi_k)$$

Matching: score and displacement

Each call to f runs **LoFTR**[6] between $\mathcal{I}$ and the render $M(x)$, then **RANSAC**[7] keeps only the matches that fit one affine transform. One call gives two things:

- **Inlier count** $f(x)$, the score we try to maximise
- the inliers' confidence-weighted **mean displacement** $d(x) = \sum_i \bar{c}_i\,(q_i - p_i) \in \mathbb{R}^2$ pixels: how far off the render is, and in which direction

The tile's real size (W, H) turns d into a **geographic correction**: $d(x) \propto \widetilde{\nabla}(x)$

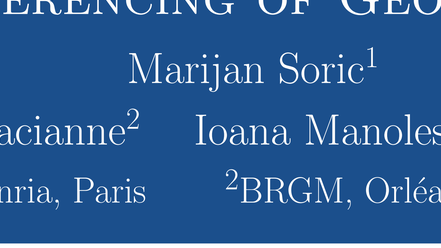


Fig. 1: LoFTR+RANSAC matches: BRGM report figure (left), OSM render (right).

We seed a dense grid around the top-3 candidates, spanning boxes 2 to 50× the size of the figure.

Pseudo-gradient $\widetilde{\nabla}$ descent

A match tells us more than how good a candidate is: if there is any overlap, it also tells us which way to move. So we stop guessing and simply go there:

$$x_{k+1} = x_k + \widetilde{\nabla}(x_k),$$

We stop when a step falls below a threshold or repeats itself. We start again from each of the **top-5 seeds**.

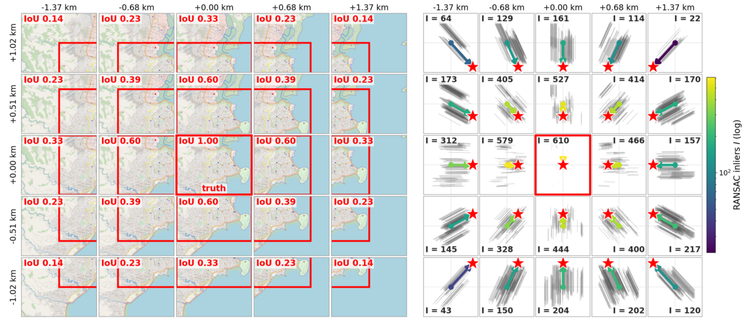


Fig. 2: *Left:* OSM render, with the true window in red. *Right:* where every matched point says to go (grey), their average $\bar{\nabla}$ (arrow), and the truth ($\star$).

πBO as explorer

When the descent gets stuck, a GP surrogate, tilted toward the prior, looks over a wider area:

$$x_{k+1} \in \arg\max_x \ \mathrm{EI}(x)\,\pi(x).$$

I is flat almost everywhere and then jumps. We use πBO[8] to bring in prior knowledge from textual data $\mathcal{T}$.

Results

dataset 2: 153 real BRGM figures (Mayotte). **dataset 1**: 971 synthetic figures (France). **IoU** compares the predicted map window with the true one.

	dataset 2, $n = 153$						**dataset 1**, $n = 971$					
	error < km			IoU >			error < km			IoU >		
method	0.1	1	5	0	0.5	median	0.1	1	5	0	0.5	median
Seed only	7	61	80	71	38	0.54 km	3	50	74	75	49	0.97 km
$\widetilde{\nabla}$	58	69	82	73	60	**0.02 km**	**57**	**74**	75	75	**73**	**0.06 km**
$\widetilde{\nabla} + \pi$BO	**59**	**71**	80	73	**62**	**0.02 km**	**57**	**74**	75	75	**73**	**0.06 km**
GPT-5.2	8	**71**	**92**	**78**	42	0.59 km	1	41	**81**	**76**	44	1.39 km

- **The LLM finds the area; our pipeline is more precise.** GPT answers every figure, even the ones we cannot attempt, and it is the best at landing in the right region. It just never lands *on* the map: 8% within 100 m, against our 58%.
- **The prior decides *if*, the descent decides *how well*.** On dataset 1, the same 729 of 971 figures overlap the truth before and after the search. What the descent adds is precision: IoU$> \frac{1}{2}$ goes from 49 to 73%.
- **πBO is worth only its rescue.** On real reports it wins 2 figures @1 km, for 60% more renders. On synthetic data it moves 50 of 971 answers (21 better, 29 worse) and the totals do not change.

Errors and next steps

On dataset 2, figures we still miss at 1 km fail for one of three reasons: no place name we can geocode, so we never even start; a seed that lands too far away; or a cross-modal matching failure, often a topographic or aerial figure against an OSM render. **Next steps:**

- Georeference a report's figures *together*, not one at a time, using their similarity $(\mathcal{I}, \mathcal{T})$
- Use GPT-5.2 as a seed
- Compute a confidence score over the whole pipeline to get calibrated scores

References

[1] Duan, W. et al. (2025). DIGMAPPER: A Modular System for Automated Geologic Map Digitization. *SIGSPATIAL*, 717–728.

[2] Hackeloeer, A. et al. (2014). Georeferencing: A Review of Methods and Applications. *Annals of GIS*, 20(1), 61–69.

[3] Vaienti, B. et al. (2026). Georeferencing Historical Maps Using Local Feature Matching and Delaunay Consistency. *Cartography and Geographic Information Science*, 1–23.

[4] Auer, C. et al. (2024). Docling Technical Report. *arXiv:2408.09869*.

[5] Kim, J. et al. (2023). The MapKurator System: A Complete Pipeline for Extracting and Linking Text From Historical Maps. *SIGSPATIAL*.

[6] Sun, J. et al. (2021). LoFTR: Detector-Free Local Feature Matching With Transformers. *CVPR*.

[7] Fischler, M. and Bolles, R. (1981). Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6), 381–395.

[8] Hvarfner, C. et al. (2022). πBO: Augmenting Acquisition Functions With User Beliefs for Bayesian Optimization. *ICLR*.

[9] OpenStreetMap contributors (2026). *OpenStreetMap*. https://www.openstreetmap.org.