

Using textureless, low-detailed 3D city models for visual localization

Yasmin Loeper¹, Markus Gerke¹, Phillipp Fanta-Jende²

¹ Institute of Geodesy and Photogrammetry, Technische Universität Braunschweig,
Brunswick, Germany - (y.loeper, m.gerke)@tu-braunschweig.de

² Unit Assistive and Autonomous Systems, Center for Vision, Automation and Control,
AIT Austrian Institute of Technology, Vienna, Austria - phillipp.fanta-jende@ait.ac.at

Keywords: Visual localization, CityGML, 3D city models, feature matching

Abstract

Accurate camera pose estimation in urban environments remains challenging when reference imagery is generated from low-detailed, textureless 3D city models and must be matched against real world imagery. In this work we (i) extend our existing iterative object-based visual localization approach with an additional semantic feature and (ii) conduct a systematic baseline study of contemporary feature matching techniques on such cross-domain query-reference image pairs. As a first step to close the domain gap, we augment our iterative object-based visual localization pipeline with semantic masks derived from a pretrained semantic segmentation model. Intersection-over-Union between query and rendered masks is incorporated into the matching score, leading to a better pose accuracy. For the baseline study, we use a range of feature matching techniques: handcrafted (SIFT, AKAZE, ORB, FAST), learned detectors (XFeat, Key.Net, DeDoDe, DISK, AffNet), learned descriptors (XFeat, DISK, DeDoDe, HardNet), learned matchers (LightGlue, LoFTR), the line matcher SOLD2, and the learned matchers MINIMA-RoMa, MINIMA-LoFTR, MINIMA-XoFTR, and MatchAnything, which were trained on cross-modality datasets. The cross-modality focused matchers achieved the best results. For 20% 10%, 9%, and 7% of the evaluated query images the estimated camera pose had a translation error less than 5 m and a rotation error less than 5°. In this context, the other methods were only able to achieve a maximum success rate of 1.4%.

1. Introduction

Usually, absolute position estimation is based on GNSS. However, in urban environments, the location derived from GNSS may be inaccurate. One reason is, for example, multipath effects. Visual localization, in contrast, offers an alternative or complementary component to the usual GNSS positioning. Visual localization finds applications across a wide array of fields. In robotics (Maggio et al., 2022), Unmanned Aerial Vehicle (UAV) navigation (Wu et al., 2024), autonomous driving (Heng et al., 2018), and augmented reality (Middelberg et al., 2014), visual localization was successfully used for pose estimation.

To estimate the camera pose, a query image is compared with reference data. In the field of visual localization the reference data is often referred to as map representation. The map representation can be database images with known ground truth poses (Sarlin et al., 2021), a Structure-from-Motion (SfM) point cloud (Schönberger et al., 2018) or a 3D mesh (Panek et al., 2022, Vultaggio et al., 2024). An SfM point cloud requires images with known poses and pre-processing steps. In addition, SfM demands a lot of memory. 3D meshes are also based on pre-processing steps and georeferenced images. However, the availability of these map representations limits the scalability and public access to visual localization. This could inhibit further development and application of visual localization.

The question arises whether there is reference data that is already available for many cities worldwide. 3D city models, in particular provided as City Geography Markup Language (CityGML)¹ data, are an example of this. The models are not memory-intensive and are georeferenced. However, employing 3D city models for visual localization presents a challenge. 3D city

models are usually textureless, colorless, and because of generalization they may not be of high geometric detail.

Although textureless, colorless, and low-detail 3D city models are a difficult data source, they are a promising way to represent the map. Another problem with current approaches that use textureless 3D models is the precision of the ground truth pose. Many datasets provide ground truth poses that are defined relative to the 3D model of the map rather than in an absolute coordinate system. Any statement about the accuracy of the approaches should therefore be treated with caution.

In our previous work, we demonstrated an approach that uses 3D city models for visual localization (Loeper et al., 2024). The results show that relying solely on geometric line features is not sufficient. A Level of Detail (LoD) 2 city model does not contain detailed information about the geometric appearance of buildings, e.g. windows or doors. Consequently, the mismatch between the geometric features in the query image and those in the reference images creates ambiguities. Lines originating from windows or doors are often matched to building outlines, leading to large translational errors.

Since the issues mentioned can be traced back to the discrepancy between the amount of visual content and the richness of features in the reference image versus the query image, learned feature detectors, descriptors, and matchers are a possible way to bridge this gap. We investigate whether existing methods already provide a basis for solving the problem of cross-domain matching between real and rendered images of 3D city models.

2. Related Work

We define the use of textureless and low-detailed 3D models in visual localization as an object-based approach. By object-

¹ <https://www.ogc.org/standard/citygml/>

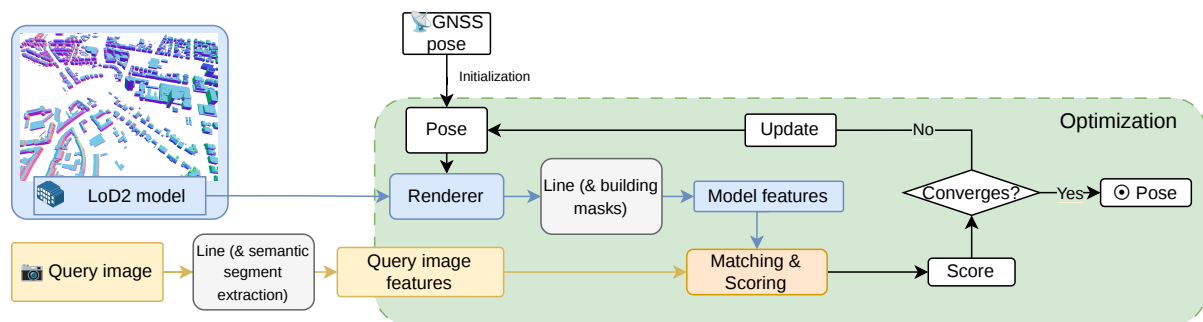


Figure 1. Overall workflow of the object-based visual localization approach, based on (Loeper et al., 2024).

based visual localisation, we mean the exploitation of geometric characteristics of an object as features for matching for pose estimation. The approaches for object-based visual localization can be compared with those for mesh-based approaches. They can be roughly divided into hierarchical, scene and pose regression methods, and iterative methods. Hierarchical methods are also used in combination with regression methods.

In hierarchical approaches, 2D features in a query image are matched against either real reference views or, when none exist, synthetic renders of the model. Those 2D–2D correspondences are then lifted into 3D to calculate the pose (Panek et al., 2023). Iterative methods instead generate pose candidates, render the model under each hypothesis, and refine the pose by aligning query-image features with non-photorealistic renderings (Zhu et al., 2024). A major challenge across both pipelines is reliable cross-domain matching between real photographs and textureless renders, which calls for bespoke feature-matching techniques. An alternative strategy bypasses explicit matching by training neural networks directly on synthetic images, or on edge and segmentation maps derived from them, with known ground-truth poses (Acharya et al., 2022). Although some approaches add textures or photorealistic detail before extracting training features, the direct use of low-detail, textureless models for pose regression remains largely unexplored.

A central problem in visual localization is image matching, i.e., generating 2D–2D pixel correspondences between a query image and a reference image. Image matching can be divided into several stages: feature detection, feature description, and matching. Under handcrafted methods, both detectors and descriptors are typically based on image gradients (Lowe, 2004, Alcantarilla et al., 2013, Rublee et al., 2011). Descriptors may, for example, be vectors of orientation histograms such as SIFT (Lowe, 2004). Matching is then performed using a nearest-neighbor (NN) search. Learning-based approaches follow a similar pipeline but rely for example on convolutional neural networks (CNNs) to learn keypoint detectors (DeTone et al., 2017, Dusmanu et al., 2019, Potje et al., 2024, Tyszkiewicz et al., 2020) and descriptors (Potje et al., 2024, Tian et al., 2019). Matching is again based on NN or mutual nearest-neighbor (MNN) criteria, after which outlier rejection is carried out with ratio tests, inlier classifiers, either learned (Zhang et al., 2022) or handcrafted (RANSAC (Fischler and Bolles, 1981)), and geometric models. Recent deep matcher approaches, which are learning-based, often employ graph neural networks (GNNs) or CNNs to replace the final two steps of the pipeline (Lindenberg et al., 2023, Sun et al., 2021, Sarlin et al., 2019). These deep matchers can be further categorized as sparse matchers (Sarlin et al., 2019, Lindenberg et al., 2023), dense matchers (Edstedt et al., 2023), and semi-dense

matchers (Potje et al., 2024, Sun et al., 2021).

In urban environments, straight lines provide an alternative feature for image matching. Similar to point-based methods, line detectors can be handcrafted (Akinlar and Topal, 2011, Grompone von Gioi et al., 2012, Suárez et al., 2022, Fernandes and Oliveira, 2008), learning-based (Huang et al., 2018, Zhou et al., 2019, Dai et al., 2021, Pautrat et al., 2021), or hybrid (Pautrat et al., 2022, Teplyakov et al., 2022). A key advantage of line-based matching is the ability to exploit the structural relationships of lines. For example, the region surrounding a line can be used to compute a descriptor (Lange et al., 2020). Other line-matching strategies sample points along the line and use those points for matching (Pautrat et al., 2021) or apply GNNs (Ubungazhibov et al., 2025).

In recent years, several cross-modality matching methods have been introduced. These approaches enable feature matching between two different modalities in comparison to the previously mentioned techniques which are specialized for a single domain. Some methods bridge this gap by extending the methods with cross-modality training data (He et al., 2025), while others aim to generalize the matching process itself (Jiang et al., 2024). XoFTR (Tuzcuoğlu et al., 2024) is specifically designed for matching thermal infrared images with RGB images. OmniGlue (Jiang et al., 2024), on the other hand, augments traditional sparse matchers with dense guidance features. ReDFeat (Deng and Ma, 2022) jointly trains detectors and descriptors using a coupled loss, encouraging the keypoints to be maximally discriminative for matching. MatchAnything (He et al., 2025) extends existing matching frameworks such as ELoFTR (Wang et al., 2024) with a pre-training framework. A comparable approach is MINIMA (Ren et al., 2024), which curates a new multimodal dataset and uses it to train established matchers like LoFTR (Sun et al., 2021), XoFTR (Tuzcuoğlu et al., 2024), and RoMa (Edstedt et al., 2023). Both strategies suggest that the choice of training data has a greater impact on cross-modal matching accuracy than the specific network architecture.

3. Methodology

We consider two mutually independent approaches: one to enrich the iterative object-based visual localization approach with an additional feature and another to evaluate various feature matching methods. The iterative object-based visual localization approach is explained in section 3.1. The methodology for the feature matching methods, i.e. the baseline study, is described in section 3.2.

3.1 Iterative object-based visual localization approach

We implement an iterative scheme comparable to the approach presented in our previous work (Loeper et al., 2024). The original pipeline is extended with an additional feature extractor, semantic segmentation, and this extra modality is incorporated into the iterative matching loop to improve robustness.

Our workflow is illustrated in Figure 1. The workflow consists of four modules: the rendering, feature extraction, matching and scoring, and optimization module. During the optimization, the pose is used to render the corresponding scene of the city model. Next, the lines are detected and extracted from the rendered image (marked in blue). Similarly, lines are also detected and extracted from the query image (marked in yellow). As soon as the lines of the query image and those of the rendering have been extracted, they are forwarded to the matching and scoring module. Lines are compared on the basis of their geometric features such as direction, position and length. The score is calculated based on the match of the geometry.

In addition to line features, semantic features are employed as further basis for scoring and matching. A semantic segmentation model is used to identify and aggregate the relevant areas in the query image that will participate in the scoring process. Which semantic classes are considered relevant depends on the segmentation model, but in practice they can be defined as all regions that describe a building. This yields a binary mask. The mask can be directly compared with the rendered image, because the building-only rendering already provides a natural mask: the rendered building surfaces are the only pixels that carry non-zero color values. The consistency between the two masks is quantified by the Intersection-over-Union (IoU) measure, and the resulting IoU score is incorporated into the overall matching score. The best score and the corresponding pose is the result of the optimization process.

3.2 Baseline study

Our method for the baseline study consists of a simplified visual localization workflow. The spatial-retrieval component forms the first step. Using the ground-truth pose of the query image, we retrieve $k-1$ reference images that lie within a 30 m radius and the reference image that has the best viewing direction, obtained via a KD-Tree search. By ensuring that the best-viewing-direction image is included, we guarantee that at least one reference image exhibits a high visual overlap with the query. This component simulates an image-retrieval process that relies on a pose prior, for example provided by GNSS. Even when GNSS signals suffer from occlusion or multipath, a practical workflow can still rely on a pose prior for spatial retrieval. The GNSS solution will typically be accurate within a 30 m to 40 m radius, or the trajectory can be approximated from previously known positions, providing a sufficiently good initial estimate for the retrieval stage.

In the second step, feature matching is performed. Each of the k reference images is matched against the query image. The exact procedure depends on the chosen feature matcher. Detect-and-describe approaches such as SIFT, ORB (Rublee et al., 2011), AKAZE (Alcantarilla et al., 2013), and XFeat (Potje et al., 2024) extract keypoints and descriptors from the reference and query images independently, after which the matches are filtered with a ratio test or brute-force matching. End-to-end methods such as MINIMA-LOFTR, LoFTR, and SOLD2 (Pautrat et al., 2021) process both images jointly and directly output the matched points. The result of this step

is a set of raw matches, i.e., pairs of 2D pixel coordinates $(x_{\text{ref}}, y_{\text{ref}}), (x_{\text{query}}, y_{\text{query}})$.

The third step is geometric verification. Because the raw matches still contain outliers, we apply RANSAC to estimate a fundamental matrix F and to classify each match as an inlier or an outlier.

In the fourth step, the inliers are projected into 3D. For each inlier keypoint we obtain its depth from the depth map of the reference image at the keypoint location (u, v) : $d = \text{depthmap}[u, v]$. Using the depth value we back-project the keypoint to camera coordinates via

$$p_{\text{cam}} = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot d.$$

At the same time, with the known ground-truth poses of the query images we compute the reprojection error.

The final step of the visual-localization pipeline uses the N 2D–3D correspondences to compute the query camera pose with a Perspective-n-Point (PnP) algorithm. Two variants are evaluated: a single-reference variant that employs only the 2D–3D pairs from one reference image, and a multi-reference variant that aggregates the correspondences from all k reference images.

4. Experiments

4.1 Dataset

We test the feature matching methods and our approach in a $370 \times 280 \text{m}^2$ area near Braunschweig's city center in Lower Saxony in Germany, featuring diverse buildings and urban canyons.

We use the Level of Detail (LoD)2 Braunschweig CityGML data, built from cadastral outlines, a 5 m digital terrain model (DTM) and laser/point cloud data. Therefore, the horizontal accuracy follows the cadastral data and vertical accuracy the point cloud. Our query dataset comprises 960×1280 images in 4 cm ground sampling distance (GSD) from a smartphone mounted on a tactical grade inertial navigation system (INS), with Post-Processed Kinematic (PPK) trajectories accurate to a few centimeters. A subsequent SfM and bundle-adjustment with GNSS Real-Time Kinematic (RTK) ground control points (GCPs) and check points (CP) yielded at GCP of (6; 7; 2) cm in X, Y, Z, respectively and mean errors at CPs of (7; 4; 1) cm, mean image exterior orientation errors of 1.5 cm in X, Y and 0.7 cm in Z and standard deviations of 6 and 2 mm, and mean rotation errors of 0.02° at $\sigma = 0.04^\circ$. The CityGML data, a subset of non-covisible query images, and a textured 3D aerial mesh have been published as the egeioussBench benchmark dataset (Fanta-Jende et al., 2025). However, the test dataset we use for the baseline study and for testing the iterative object-based visual localization approach consists of more query images than the egeioussBench dataset.

Iterative object-based visual localization approach It should be noted that we did not use pre-rendered reference images for our experiments with the iterative object-based visual localization approach. The reference images for this approach

Table 1. Overview of the tested methods in our baseline study. Methods are grouped by category: **HC** (handcrafted), **LS** (learned sparse), **D+M** (detect + match), **DN** (dense), **LN** (line segments), **CM** (cross-modal).

	Method	Detector	Descriptor	Matcher
HC	SIFT	SIFT DoG	SIFT 128-d	NN, Ratio test, mutual check
	ORB	FAST	ORB binary	NN, Ratio test, mutual check
	AKAZE	AKAZE nonlinear	AKAZE binary	NN, Ratio test, mutual check
LS	XFeat	XFeat learned	XFeat 64-d	NN, Ratio test, mutual check
	XFeat*	XFeat learned	XFeat 64-d	MNN, ratio test
	XFeat-match	XFeat learned	XFeat 64-d	MNN
D+M	SIFT+LG	SIFT DoG	SIFT 128-d	LightGlue
	DISK+LG	DISK learned	DISK 128-d	LightGlue
	DeDoDe+LG	DeDoDe learned	DeDoDe 256-d	LightGlue
	KeyNet+LG	KeyNet+AffNet	HardNet 128-d	LightGlue
DN	LoFTR	none, dense	implicit	LoFTR attention
LN	SOLD2	line segments	dense desc	Line matching
CM	MatchAnything	none	ELoFTR (RepVGG)	Dual Softmax (C2F)
	MINIMA-LoFTR	none	ResNet FPN CNN	LoFTR transformer + dual softmax
	MINIMA-RoMa	none	ResNet CNN	GP decoder + ConvRefiner
	MINIMA-XoFTR	none	ResNet CNN	XoFTR transformer + dual softmax

are generated during the optimization process, and the poses for the reference images are generated based on Bayesian optimization. The query image set contains 2331 for these experiments.

Baseline study The test dataset for the baseline study comprises reference images and query images. The reference images were generated with a custom OpenGL renderer. For this purpose, the CityGML model was converted into a mesh that can be rendered. Using a sampling scheme, positions were placed along the streets of the test area at intervals of approximately 5 m to 6 m. For each sampled position, a camera pose was computed at 10° increments. The query set consists of 2709 images together with their ground-truth poses.

4.2 Iterative object-based visual localization approach

In this approach, we use a spatial bounding box within which the camera poses can be computed. The spatial bounding box is calculated based on a GNSS prior of the query poses and has a size of 40 × 40m.

For the experiments, we tested two different feature extraction and matching approaches. On the one hand, we only considered straight lines as features. For this, we implemented ELSSED (Suárez et al., 2022) as the line detector. In the matching process, we finally calculated the correspondence of the geometry of the detected lines of the query image and the rendered image for the score. In addition to the straight lines, the second approach also contains the segments from semantic segmentation. To do this, we applied the Revision-based Transformer Facade Parsing (RTFP) (Wang et al., 2023) pipeline to the query image.

4.3 Baseline study

Our baseline study selects a representative subset from the various feature matching categories. The selection of feature matching methods is shown in Table 1. To represent handcrafted methods we evaluate SIFT, AKAZE and ORB. For the learned detectors, we include XFeat, KeyNet (Barroso-Laguna et al., 2019), DeDoDe (Edstedt et al., 2024), DISK (Tyszkiewicz et al., 2020), FAST (Rosten et al., 2010) and AffNet (Mishkin et al., 2017), while the learned descriptors under test are XFeat, DISK (Tyszkiewicz et al., 2020), DeDoDe (Edstedt et al., 2024) and HardNet (Mishchuk et al., 2017). In addition, LightGlue (Lindenberger et al., 2023) is used as a learned matcher in several configurations (see

Table 1), LoFTR and XFeat-match serve as a dense end-to-end matcher, XFeat* serves as sparse end-to-end matcher, and SOLD2 is incorporated as a line-matching approach. As matching methods that were trained using cross-modality datasets, we implemented the combinations MINIMA-XoFTR, MINIMA-LoFTR, MINIMA-RoMa, and MatchAnything. These approaches are also detector-free dense matching methods.

All handcrafted techniques are implemented with OpenCV. For learning-based approaches, we employ publicly available pre-trained models. Their implementation relies on PyTorch and through the kornia library. Alternatively, if the methods were not available in these libraries, we implemented a wrapper to load the models and weights. For detect-and-describe pipelines, we apply the standard ratio test, and we perform inlier verification by estimating the fundamental matrix with RANSAC, also using OpenCV. Pose estimation is carried out with the PnP algorithm provided by OpenCV.

To keep the comparison fair, we also adopt a PnP-based solution for SOLD2. For each matched line pair (l_{ref} , l_{query}) we sample five equidistant points at parameter values $t \in \{0, 0.25, 0.5, 0.75, 1.0\}$:

$$p(t) = p_{start} + t \cdot (p_{end} - p_{start}).$$

This conversion transforms M line matches into $5 \times M$ point correspondences, which can then be processed by the standard PnP pipeline.

For the selection of the k reference images we use a KD-Tree nearest-neighbor search on the ground truth poses. Simultaneously, we enforce a viewing-direction check to guarantee that at least one of the retrieved references has an optimal viewing direction. The maximum number of reference images is fixed to 15, and only query images that obtain more than two reference images after the spatial-retrieval stage are retained for evaluation.

The evaluation follows the conventional metrics used in visual localization and feature matching studies. For each method, for every query–reference image pair, we compute the raw number of matches, the number of inliers, the inlier-to-match ratio,

Table 2. Localization results for the 2331 query images. We report the % of query images localized within the given deviation of the ground truth pose.

Feature type	2 m, 2°	5 m, 5°	9 m, 10°	15 m, 15°
L	0	0.098	0.441	1.471
L & S	0.388	2.064	5.524	7.769

Table 3. Mean absolute error (MAE) for translation and rotation, and mean absolute deviation (MAD) for the 2331 query images.

Feature type	MAE (m / °)	MAD (m / °)
L	48.834 / 39.232	17.876 / 24.804
L & S	42.461 / 42.671	19.129 / 28.387

and the median reprojection error. Aggregating over all reference images for a given query, we report the total number of inliers and the combined PnP solution. Finally, across the entire query set we report (i) the average number of inliers per query–reference pair, (ii) the average number of inliers for the pair with the best viewing-direction alignment, (iii) the average reprojection error, (iv) the count of queries for which PnP yields a translation and rotation error of less than 5 m and 5°, and (v) the mean and median translation and rotation errors of all successful PnP estimations, expressed in meters and degrees, respectively.

4.4 Results

Accuracy of iterative object-based approach To investigate the accuracy of our method, we performed the optimization of the camera pose parameters for 2331 images. The preliminary results of the tested approaches are shown in Table 2 and 3. The feature type describes the selected method as outlined in section 4 for the feature extraction. The approach with straight lines as feature extractor is labeled L. The approach with straight lines and segments from the semantic segmentation is abbreviated as L & S. Table 2 shows that the combination of line features and semantic segments led to better results. The mean absolute error (MAE) of the translation of the line approach is 48.834 m and of the rotation 39.232°. Whereas the MAE of the translation of the combined approach is 42.461 m and the rotation is 42.671°. However, the improvement of the MAE and the MAD is not as strong as would be expected.

Baseline evaluation According to section 4.3, the results are calculated per method for each query-reference image pair, per query, and aggregated over all queries. Table 4 summarizes the

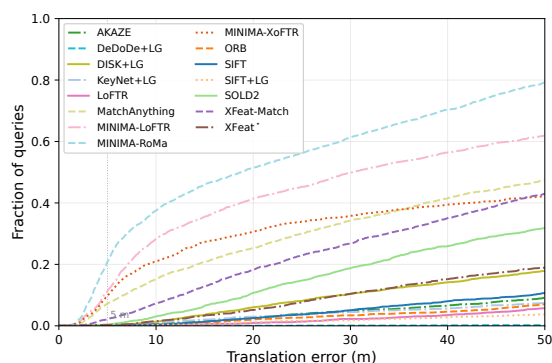


Figure 2. Cumulative distribution function of the translation error per tested method.

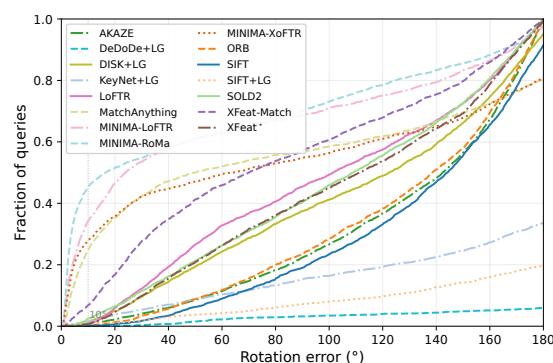


Figure 3. Cumulative distribution function of the rotation error per tested method.

results for each method aggregated over all queries. The average number of inliers per reference image is abbreviated as Inl/ref. The BestDir column reports the number of inliers obtained from the reference image that has the best viewing direction. The reprojection error is computed as described in 4.3. Translation and rotation error are expressed in meters and degrees, respectively. The PnP rate (in %) indicates the proportion of queries in which a PnP solution could be computed given a sufficient number of inliers. Finally, the success rate denote the percentage of queries whose pose estimation yielded to a translation error <5 m and a rotation error <5°. The Figure 2 shows the cumulative distribution function of the tested methods of the translation error and Figure 3 shows the cumulative distribution function of the rotation error.

MINIMA-RoMa and SOLD2 stand out because they produce the largest numbers of inliers per reference image. As expected, the inlier count for every method increases when the reference image with the viewing direction most similar to that of the query image is selected. The growth of MINIMA-RoMa, MatchAnything, MINIMA-XoFTR, and MINIMA-LoFTR is particularly noteworthy here. With a success rate of approximately 20.38% and a median translation error of 18.9 m and rotation error of 16.5°, MINIMA-RoMa achieves the best overall result. The success rates, as well as the median translation and rotation errors, of MINIMA-LoFTR, MatchAnything, and MINIMA-XoFTR, i.e. the cross-modal trained matching methods, are also worth highlighting. However, the comparatively high mean translation and rotation error suggest that the results still vary widely. Because MINIMA-RoMa was trained on cross-modal training datasets, this outcome was expected. Nevertheless, the average number of inliers per reference image cannot be used unconditionally as a measure of suitability, since the comparison of MatchAnything, MINIMA-LoFTR, and MINIMA-XoFTR with SOLD2 or XFeat-Match shows that the quality of the matches is the decisive factor. MINIMA-RoMa could represent a combination of both of these factors, as the higher total number of inliers per reference image and the potentially more accurate correspondences in the geometric and PnP pipelines reinforce each other. Rather, the significant improvement in the number of inliers per reference image for the best viewing direction could serve as an indicator of the quality of the matches. As expected, high translation and rotation errors were observed in the handcrafted methods, as well as in methods that were not explicitly trained for cross-modality matching. Among these, XFeat-Match stands out. However, the low success rate and large mean translation and rotation error indicate that a clear pattern is not discernible here.



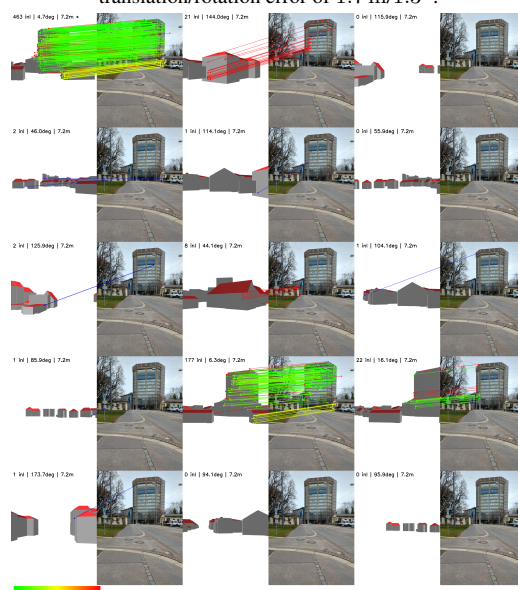
(a) MINIMA-RoMa: the estimated pose has a translation/rotation error of 1.6 m/3.1°.



(b) MINIMA-LoFTR: the estimated pose has a translation/rotation error of 1.7 m/3.3°.



(c) MINIMA-XoFTR: the estimated pose has a translation/rotation error of 2.2 m/1.2°.



(d) MatchAnything: the estimated pose has a translation/rotation error of 4.2 m/3.4°.

Figure 4. Matches for the cross-modal trained methods from the best query image and the reference images. Blue matches indicate that there was no valid depth for this match. The number of inliers and the deviation between the reference image's camera pose and the ground truth query image camera pose (m/°) are displayed in the upper-left corner of the reference image.

Table 4. Baseline matcher evaluation summary. Methods are grouped by category: **HC** (handcrafted), **LS** (learned sparse), **D+M** (detect + match), **DN** (dense), **LN** (line segments), **CM** (cross-modal). **Inl/Ref**: the average number of inliers per reference image across all queries. **BestDir**: the average number of inliers determined from the reference image with the best viewing direction. **Reproj**: the mean reprojection error. **Med Trans** and **Med Rot**: median translation and rotation errors over all queries. **Mean Trans** and **Mean Rot**: mean translation and rotation errors over all queries. **PnP**: the percentage of query images for which a PnP solution could be computed successfully. **Succ.**: the percentage of query images whose translation and rotation errors are both smaller than 5 m and 5°, respectively. The top results are highlighted in bold, and the second-best results are underlined.

	Method	Inl/ref	BestDir	Reproj (px)	Med Trans (m)	Med Rot (°)	Mean Trans (m)	Mean Rot (°)	PnP (%)	Succ. (%)
HC	SIFT	3.4	4.7	369.2	243.8	138.2	1.17e15	127.3	91.5	0.0000
	ORB	11.2	14.7	399.0	424.2	138.5	7054.5	125.5	100.0	0.0000
	AKAZE	9.3	11.8	396.0	333.1	142.4	6.81e12	127.2	100.0	0.0369
LS	XFeat	0.0	0.1	–	–	–	–	–	0.0	0.0000
	XFeat-Match	14.0	17.2	477.6	61.6	70.1	1.82e12	81.7	100.0	1.3658
	XFeat*	4.8	6.8	375.0	149.7	110.7	1.34e13	105.1	<u>99.3</u>	0.0369
D+M	SIFT+LG	0.8	1.4	358.3	213.3	121.3	5.52e12	108.4	19.7	0.0369
	DISK+LG	3.5	4.9	392.9	165.0	116.5	2100.3	107.1	95.2	0.0369
	DeDoDe+LG	0.3	0.4	512.0	455.1	84.1	1519.1	93.8	6.0	0.0000
	KeyNet+LG	1.4	2.5	<u>348.8</u>	147.7	102.1	5.11e8	98.9	33.7	0.0738
DN	LoFTR	7.0	9.2	413.7	258.2	101.5	843.9	99.9	100.0	0.0000
LN	SOLD2	<u>23.1</u>	<u>26.7</u>	484.9	89.0	108.8	453.7	103.8	100.0	0.0738
CM	MatchAnything	3.8	13.7	254.5	38.1	26.8	<u>193.6</u>	57.3	80.5	7.0137
	MINIMA-LoFTR	6.4	16.9	367.8	28.8	22.6	656.1	56.4	97.6	<u>10.521</u>
	MINIMA-RoMa	851.2	1402.4	481.5	18.9	16.5	32.2	55.0	100.0	20.377
	MINIMA-XoFTR	5.5	15.9	359.1	43.5	26.6	5481.7	59.6	80.7	9.7453

We have visualized the best query images for the cross-modal trained methods in Figure 4. The best query image, that is, the one that has the most accurate pose estimate. In addition to the matches, the visualizations encode the reprojection error using a color map. Figure 4a shows the matches for MINIMA-RoMa. Compared to the other methods, it is visible that MINIMA-RoMa has significantly more matches. The Figure 4b shows the matches for MINIMA-LoFTR for the best query image. It is evident that the number of inliers increases when the viewing direction more closely aligns with the ground truth viewing direction of the query image. Figure 4c visualizes the matches produced by MINIMA-XoFTR for the best query image. As in the previous analysis, it is evident that smaller deviations between the viewing direction of the reference image and the ground truth viewing direction of the query image lead to a larger number of estimated inliers, which is also reflected in the corresponding Table 4. The Figure 4d displays the matches generated by MatchAnything for its best query image. Again, the best viewing direction results in a substantially higher inlier count. Overall, it can be observed that the methods perform especially well on prominent structures such as the tall building visible in Figure 4b, 4c and 4d. The correlation between an increasing number of inliers and a better overlap between the query image scene and the reference image indicates that the methods are effectively detecting those structures.

The results presented in Table 4 and Figure 4 indicate that an uneven distribution of inliers across the reference images for each query image positively influences the PnP computation. More precisely, a larger number of inliers in those reference images whose viewing direction deviates less from that of the query image yields a more accurate pose estimate. To investigate this effect, we computed the translation and rotation errors as a function of the coefficient of variation of the inlier count per reference image for each query, focusing on the cross-modal

trained methods. The outcome is visualized in Figure 5. For the cross-modal approaches, a more heterogeneous inlier distribution correlates with lower error in the estimated pose. In Figure 6 we show the normalized median number of inliers versus the angular deviation between the viewing direction of a reference image and that of the query image. Because MINIMA-RoMa produces a substantially higher absolute number of inliers than the other methods, we normalized the inlier counts to enable fair comparison. One explanation for these observations is the positive discriminative capability of the matchers: they are able to reliably identify reference images that depict scene similar to the query image. In this context, an uneven distribution of inliers across the reference images is desirable, as it reflects the presence of high-quality matches that drive more precise PnP results.

5. Conclusion

In this paper, we address two problems. First, we extend our previous iterative object-based visual localization approach with an additional feature and evaluate the resulting workflow. Second, we investigate the feasibility of applying established feature matching techniques to challenging query-reference image pairs.

By extending the matching and scoring pipeline of our object-based visual localization approach with the semantic features in addition to line features, we demonstrate that pose estimation can be improved through richer feature alignment. However, the results also show that the improvement achieved is not yet sufficient.

Because the 3D city models are low-detailed and textureless, the results of the baseline study cannot be directly compared with matches obtained between textured or real world scenes. Nevertheless, our results show that many methods

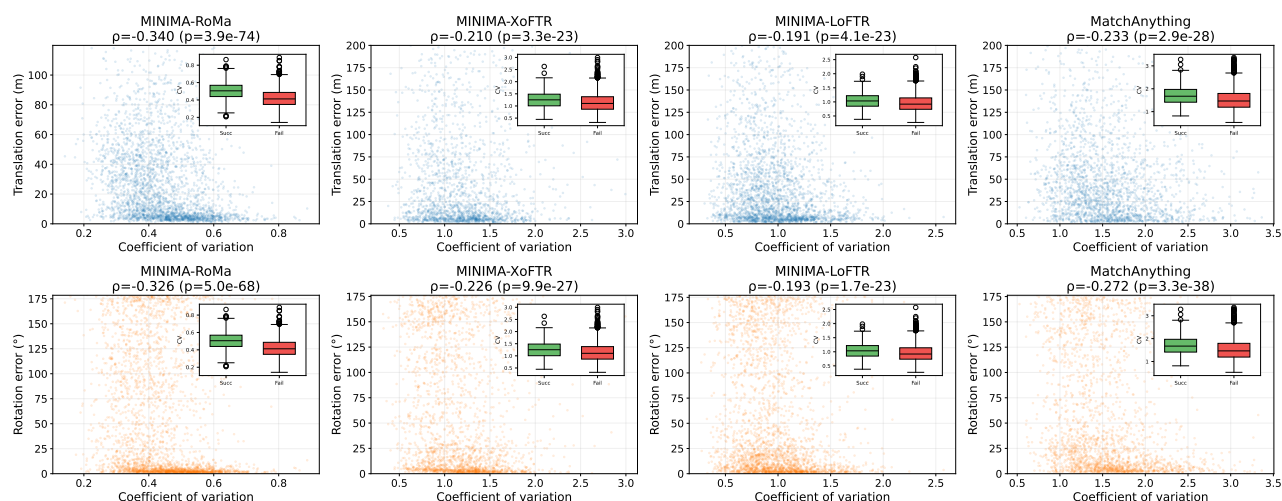


Figure 5. Coefficient of variation (CV) of per-reference inlier counts versus translation error (top row) and rotation error (bottom row) for the four cross-modal trained methods. Inset box plots compare CV distributions between successful (green, $<5\text{ m}/<5^\circ$) and failed (red) queries. Negative Spearman ρ values indicate that more uneven inlier distribution correlate with lower pose error.

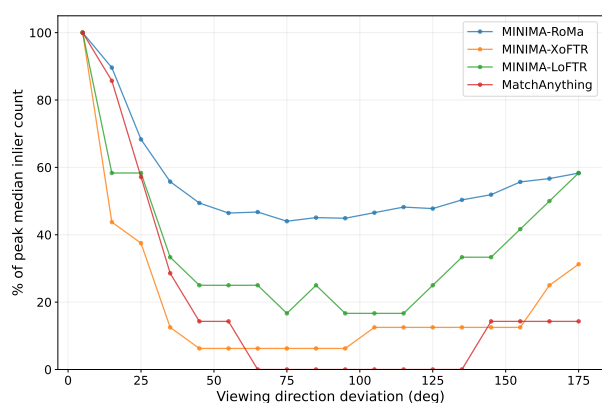


Figure 6. Normalized median inlier count as a function of viewing direction deviation between query and reference image, binned at 10° intervals. Each method's median is scaled to it's respective peak (100%).

are already capable of estimating the camera pose successfully. MINIMA-RoMa, MINIMA-LoFTR, MatchAnything, and MINIMA-XoFTR, in particular, stand out in our experiments. The comparison between our iterative object-based visual localization approach and the cross-modal trained matching methods reveals that relying solely on the geometric properties of lines and on semantic features is insufficient as a matching basis. In contrast, matchers that have been trained on cross-modal datasets provide a more robust baseline and could serve as a valuable starting point for further development. Since 3D city models continue to represent a promising reference dataset, we present this investigation as a catalyst for further research aimed at solving the cross-domain matching problem. The next step is to examine whether an extension or a custom procedure can alleviate these issues. An analysis of the preprocessing applied to query and reference images and its impact on matching quality can reveal what is required to bridge the domain gap. Possible extensions include enriching the reference renderings with additional cues, such as surface normals, edge maps, or normalized depth values, and reducing the query images to the same set of comparable attributes.

ACKNOWLEDGMENTS



Funded by
the European Union

This work is part of the EU-Horizon project *egeniouss*², which received funding under the call HORIZON-EUSPA-2021-Space with the project number 101082128.

References

- Acharya, D., Tennakoon, R., Muthu, S., Khoshelham, K., Hoseinnezhad, R., Bab-Hadiashar, A., 2022. Single-image localisation using 3D models: Combining hierarchical edge maps and semantic segmentation for domain adaptation. *Automation in Construction*, 136, 104152.
- Akinlar, C., Topal, C., 2011. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13), 1633–1642.
- Alcantarilla, P., Nuevo, J., Bartoli, A., 2013. Fast explicit diffusion for accelerated features in nonlinear scale spaces. T. Burghardt, D. Damen, W. Mayol-Cuevas, M. Mirmehdi (eds), *Proceedings of the British Machine Vision Conference 2013*, British Machine Vision Association, 13.1–13.11.
- Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K., 2019. Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters. *10.48550/arXiv.1904.00889*.
- Dai, X., Gong, H., Wu, S., Yuan, X., Ma, Y., 2021. Fully Convolutional Line Parsing. *10.48550/arXiv.2104.11207*.
- Deng, Y., Ma, J., 2022. ReDFeat: Recoupling Detection and Description for Multimodal Feature Learning. *10.48550/arXiv.2205.07439*.
- DeTone, D., Malisiewicz, T., Rabinovich, A., 2017. SuperPoint: Self-Supervised Interest Point Detection and Description. *https://arxiv.org/pdf/1712.07629*.

² <https://www.egeniouss.eu/>

- Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T., 2019. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. <https://arxiv.org/pdf/1905.03561>.
- Edstedt, J., Bökman, G., Zhao, Z., 2024. DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector. [10.48550/arXiv.2404.08928](https://arxiv.org/abs/2404.08928).
- Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M., 2023. RoMa: Robust Dense Feature Matching. <https://arxiv.org/pdf/2305.15404>.
- Fanta-Jende, P., Vultaggio, F., Kern, A., Loeper, Y., Gerke, M., 2025. egenioussBench: A New Dataset for Geospatial Visual Localisation. XLVIII-1/W6-2025, 75–82. [10.5194/isprs-archives-XLVIII-1-W6-2025-75-2025](https://arxiv.org/abs/2501.05194).
- Fernandes, L. A., Oliveira, M. M., 2008. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition*, 41(1), 299–314.
- Fischler, M. A., Bolles, R. C., 1981. Random sample consensus. 24(6), 381–395. [10.1145/358669.358692](https://doi.org/10.1145/358669.358692).
- Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. LSD: a Line Segment Detector. *Image Processing On Line*, 2, 35–55.
- He, X., Yu, H., Peng, S., Tan, D., Shen, Z., Bao, H., Zhou, X., 2025. MatchAnything: Universal Cross-Modality Image Matching with Large-Scale Pre-Training. [10.48550/arXiv.2501.07556](https://arxiv.org/abs/2501.07556).
- Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., Liu, P., Nguyen, R., Yeo, Y. C., Geiger, A., Lee, G. H., Pollefeys, M., Sattler, T., 2018. Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System.
- Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y., 2018. Learning to Parse Wireframes in Images of Man-Made Environments. <https://arxiv.org/pdf/2007.07527v1>.
- Jiang, H., Karpur, A., Cao, B., Huang, Q., Araujo, A., 2024. OmniGlue: Generalizable Feature Matching with Foundation Model Guidance. [10.48550/arXiv.2405.12979](https://arxiv.org/abs/2405.12979).
- Lange, M., Raisch, C., Schilling, A., 2020. WLD: A Wavelet and Learning based Line Descriptor for Line Feature Matching. [10.2312/vmv.20201186](https://arxiv.org/abs/2012.2312).
- Lindenberger, P., Sarlin, P.-E., Pollefeys, M., 2023. LightGlue: Local Feature Matching at Light Speed. <https://arxiv.org/pdf/2306.13643>.
- Loeper, Y., Gerke, M., Alamouri, A., Kern, A., Bajauri, M. S., Fanta-Jende, P., 2024. Visual localization in urban environments employing 3D city models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2/W8-2024, 311–318. <https://isprs-archives.copernicus.org/articles/XLVIII-2-W8-2024/311/2024/>.
- Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. 60(2), 91–110. PII: 5273355.
- Maggio, D., Abate, M., Shi, J., Mario, C., Carlone, L., 2022. Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields. <https://arxiv.org/abs/2209.09050>.
- Middelberg, S., Sattler, T., Untzelmann, O., Kobbelt, L., 2014. Scalable 6-dof localization on mobile devices. D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds), *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, 8690, Springer International Publishing, Cham, 268–283.
- Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J., 2017. Working hard to know your neighbor's margins: Local descriptor learning loss. [10.48550/arXiv.1705.10872](https://arxiv.org/abs/1705.10872).
- Mishkin, D., Radenovic, F., Matas, J., 2017. Repeatability Is Not Enough: Learning Affine Regions via Discriminability. [10.48550/arXiv.1711.06704](https://arxiv.org/abs/1711.06704).
- Panek, V., Kukulova, Z., Sattler, T., 2022. MeshLoc: Mesh-Based Visual Localization. [10.48550/arXiv.2207.10762](https://arxiv.org/abs/2207.10762).
- Panek, V., Kukulova, Z., Sattler, T., 2023. Visual Localization using Imperfect 3D Models from the Internet. [10.48550/arXiv.2304.05947](https://arxiv.org/abs/2304.05947).
- Pautrat, R., Barath, D., Larsson, V., Oswald, M. R., Pollefeys, M., 2022. DeepLSD: Line Segment Detection and Refinement with Deep Image Gradients. [arXiv; 10.48550/arXiv.2212.07766](https://arxiv.org/abs/2212.07766).
- Pautrat, R., Lin, J.-T., Larsson, V., Oswald, M. R., Pollefeys, M., 2021. SOLD2: Self-supervised Occlusion-aware Line Description and Detection. [10.48550/arXiv.2104.03362](https://arxiv.org/abs/2104.03362).
- Potje, G., Cadar, F., Araujo, A., Martins, R., Nascimento, E. R., 2024. XFeat: Accelerated Features for Lightweight Image Matching. <https://arxiv.org/abs/2404.19174>.
- Ren, J., Jiang, X., Li, Z., Liang, D., Zhou, X., Bai, X., 2024. MINIMA: Modality Invariant Image Matching. [10.48550/arXiv.2412.19412](https://arxiv.org/abs/2412.19412).
- Rosten, E., Porter, R., Drummond, T., 2010. Faster and better: a machine learning approach to corner detection. 32(1), 105–119. Journal Article.
- Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: An efficient alternative to sift or surf. 2011 *International Conference on Computer Vision (ICCV 2011)*, Institute of Electrical and Electronics Engineers, IEEE, 2564–2571.
- Sarlin, P.-E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2019. SuperGlue: Learning Feature Matching with Graph Neural Networks. <https://arxiv.org/pdf/1911.11763>.
- Sarlin, P.-E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T., 2021. Back to the Future: Learning Robust Camera Localization from Pixels to Pose. [10.48550/arXiv.2103.09213](https://arxiv.org/abs/2103.09213).
- Schönberger, J. L., Pollefeys, M., Geiger, A., Sattler, T., 2018. Semantic visual localization. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 6896–6906.
- Suárez, I., Buenaposada, J. M., Baumela, L., 2022. ELSed: Enhanced line SEgment drawing. *Pattern Recognition*, 127, 108619.
- Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X., 2021. LoFTR: Detector-Free Local Feature Matching with Transformers. <https://arxiv.org/pdf/2104.00680>.

Teplyakov, L., Erlygin, L., Shvets, E., 2022. LSDNet: Trainable Modification of LSD Algorithm for Real-Time Line Segment Detection. *IEEE Access*, 10, 45256–45265.

Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V., 2019. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. <https://arxiv.org/pdf/1904.05019>.

Tuzcuoğlu, Ö., Köksal, A., Sofu, B., Kalkan, S., Alatan, A. A., 2024. XoFTR: Cross-modal Feature Matching Transformer. *10.48550/arXiv.2404.09692*.

Tyszkiewicz, M. J., Fua, P., Trulls, E., 2020. DISK: Learning local features with policy gradient. <https://arxiv.org/pdf/2006.13566>.

Ubingazhibov, A., Pautrat, R., Suárez, I., Liu, S., Pollefeys, M., Larsson, V., 2025. Lightgluestick: a fast and robust glue for joint point-line matching. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 7303–7313.

Vultaggio, F., Fanta-Jende, P., Schörghuber, M., Kern, A., Gerke, M., 2024. Investigating Visual Localization Using Geospatial Meshes. XLVIII-2/W8-2024, 447–454. 10.5194/isprs-archives-XLVIII-2-W8-2024-447-2024.

Wang, B., Zhang, J., Zhang, R., Li, Y., Li, L., Nakashima, Y., 2023. Improving Facade Parsing with Vision Transformers and Line Integration. <https://arxiv.org/abs/2309.15523>.

Wang, Y., He, X., Peng, S., Tan, D., Zhou, X., 2024. Efficient LoFTR: Semi-Dense Local Feature Matching with Sparse-Like Speed. <https://arxiv.org/abs/2403.04765>.

Wu, R., Cheng, X., Zhu, J., Liu, X., Zhang, M., Yan, S., 2024. UAVD4L: A Large-Scale Dataset for UAV 6-DoF Localization. <https://arxiv.org/abs/2401.05971>.

Zhang, J., Sun, D., Luo, Z., Yao, A., Chen, H., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H., 2022. OANet: Learning Two-View Correspondences and Geometry Using Order-Aware Network. 44(6), 3110–3122. Journal Article.

Zhou, Y., Qi, H., Ma, Y., 2019. End-to-End Wireframe Parsing. *10.48550/arXiv.1905.03246*.

Zhu, J., Yan, S., Wang, L., Zhang, S., Liu, Y., Zhang, M., 2024. LoD-Loc: Aerial Visual Localization using LoD 3D Map with Neural Wireframe Alignment. *10.48550/arXiv.2410.12269*.