Weakly-supervised Camera Localization by Ground-to-satellite Image Registration

Yujiao Shi[1], Hongdong Li[2], Akhil Perincherry[3], and Ankit Vora[3]

[1] ShanghaiTech University, China
[2] The Australian National University, Australia
[3] Ford Motor Company, USA
shiyj2@shanghaitech.edu.cn

1 Sensitiveness to Coarse Pose Errors

In this section, we investigate the performance of our method under varying coarse pose errors.

Range of location errors. Table A presents the performance comparison between our method and the state-of-the-art [26, 28], across different ranges of location errors: 28×28 m^2 and 56×56 m^2, while maintaining the same orientation ambiguity of 20°. The results show that our method achieves consistently the best performance on cross-area evaluation.

Table A: Performance comparison with different location error ranges on the cross-view KITTI dataset.

Location Error (m^2)	Algorithms	Test-1 (Same-area) Lateral $d=1\uparrow$	Lateral $d=3\uparrow$	Longitudinal $d=1\uparrow$	Longitudinal $d=3\uparrow$	Azimuth $\theta=1\uparrow$	Azimuth $\theta=3\uparrow$
28×28	Shi and Li [26]*	44.66	73.92	12.06	35.62	25.31	57.41
	Shi *et al.* [28]*	85.85	98.46	23.27	46.99	98.89	99.97
	Ours ($\lambda=0$)	61.46	87.76	13.44	38.14	99.76	100.00
	Ours ($\lambda=1$)	66.39	94.38	18.18	53.59	99.76	100.00
56×56	Shi and Li [26]*	35.54	70.77	5.22	15.88	19.64	51.76
	Shi *et al.* [28]*	76.44	96.34	23.54	50.57	99.10	100.00
	Ours ($\lambda=0$)	59.58	85.74	11.37	31.94	99.66	100.00
	Ours ($\lambda=1$)	66.07	94.22	16.51	49.96	99.66	100.00

Location Error (m^2)	Algorithms	**Test-2 (Cross-area)** Lateral $d=1\uparrow$	Lateral $d=3\uparrow$	Longitudinal $d=1\uparrow$	Longitudinal $d=3\uparrow$	Azimuth $\theta=1\uparrow$	Azimuth $\theta=3\uparrow$
28×28	Shi and Li [26]*	34.17	72.30	11.56	35.08	11.40	48.18
	Shi *et al.* [28]*	60.01	87.96	14.69	35.64	99.42	100.00
	Ours ($\lambda=0$)	65.62	90.32	13.46	38.53	99.97	100.00
	Ours ($\lambda=1$)	67.90	89.76	14.29	42.92	99.97	100.00
56×56	Shi and Li [26]*	27.82	59.79	5.75	16.36	18.42	49.72
	Shi *et al.* [28]*	57.72	86.77	14.15	34.59	98.98	100.00
	Ours ($\lambda=0$)	62.73	86.53	9.98	29.67	99.99	100.00
	Ours ($\lambda=1$)	64.74	86.18	11.81	34.77	99.99	100.00

Table B: Performance comparison with different location error ranges on the cross-view KITTI dataset.

Orientation Ambiguity	Algorithms	Test-1 (Same-area) Lateral $d=1\uparrow$	Lateral $d=3\uparrow$	Longitudinal $d=1\uparrow$	Longitudinal $d=3\uparrow$	Azimuth $\theta=1\uparrow$	Azimuth $\theta=3\uparrow$
20°	Shi and Li [26]*	35.54	70.77	5.22	15.88	19.64	51.76
	Shi *et al.* [28]*	76.44	96.34	23.54	50.57	99.10	100.00
	Ours ($\lambda=0$)	59.58	85.74	11.37	31.94	99.66	100.00
	Ours ($\lambda=1$)	66.07	94.22	16.51	49.96	99.66	100.00
80°	Shi and Li [26]*	26.95	62.39	5.14	15.69	3.10	8.88
	Shi *et al.* [28]*	70.21	95.47	22.29	48.90	53.27	93.98
	Ours ($\lambda=0$)	53.11	86.03	12.99	32.18	57.65	96.79
	Ours ($\lambda=1$)	57.94	91.49	17.73	47.44	57.65	96.79

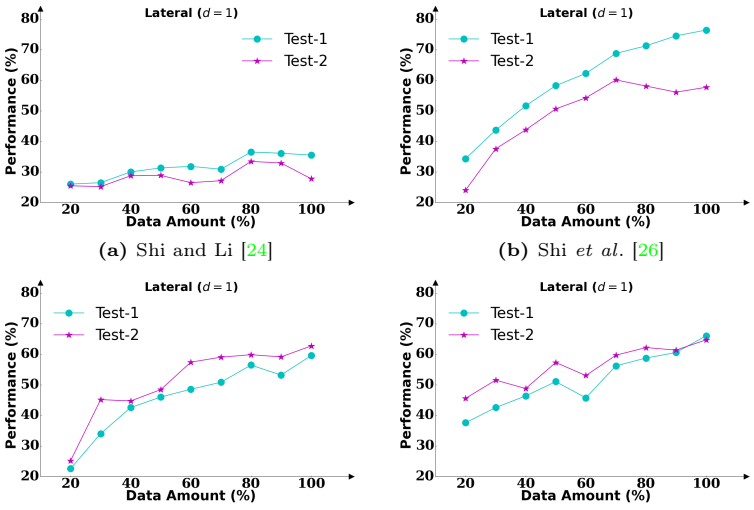
Orientation Ambiguity	Algorithms	**Test-2 (Cross-area)** Lateral $d=1\uparrow$	Lateral $d=3\uparrow$	Longitudinal $d=1\uparrow$	Longitudinal $d=3\uparrow$	Azimuth $\theta=1\uparrow$	Azimuth $\theta=3\uparrow$
20°	Shi and Li [26]*	27.82	59.79	5.75	16.36	18.42	49.72
	Shi *et al.* [28]*	57.72	86.77	14.15	34.59	98.98	100.00
	Ours ($\lambda=0$)	62.73	86.53	9.98	29.67	99.99	100.00
	Ours ($\lambda=1$)	64.74	86.18	11.81	34.77	99.99	100.00
80°	Shi and Li [26]*	22.43	54.63	5.17	15.78	3.05	8.50
	Shi *et al.* [28]*	56.97	87.72	15.17	35.39	58.68	95.92
	Ours ($\lambda=0$)	57.68	84.92	11.64	31.52	56.79	97.76
	Ours ($\lambda=1$)	60.50	86.57	12.62	35.60	56.79	97.76

Variation in orientation ambiguity. Subsequently, we augment the orientation ambiguity from 20° to 80°, while maintaining a location error range of 56×56 m^2. Table B provides the performance comparison between our method and the two state-of-the-art [26, 28]. Our method achieves the best cross-area evaluation performance on the different orientation ambiguity. Furthermore, the results reveal a decline for all methods in the percentage of images for which the estimated orientation is restricted to 1° as the orientation ambiguity increases. Nevertheless, our method and Shi *et al.* [28], which was recently accepted to ICCV2023, consistently maintain the majority of image orientations within a 3° margin from their ground truth values. Consequently, the translation estimation performance remains robust. In contrast, Shi and Li [26] encounter a notable drop in both translation and orientation estimation performance.

2 Performance with Increasing Amounts of Data as Supervision

Below, we analyze the performance of our method with $\lambda = 0, 1$ and the state-of-the-art [26, 28], when different amounts of training data are employed. The results are illustrated in Fig. A.

For most models, except Shi and Li [26], we observe a consistent increase in performance on the same-area evaluation (Test-1) as the amount of training data increases. However, when it comes to the cross-area evaluation (Test-2),



(a) Shi and Li [24] **(b)** Shi *et al.* [26]

(c) Ours ($\lambda = 0$) **(d)** Ours ($\lambda = 1$)

Fig. A: Performance comparison between our method and the state-of-the-art on the KITTI dataset with different amounts of training data.

the two state-of-the-art methods, which require ground truth pose for supervision, exhibit a decline in performance when the training data exceeds 80%. This phenomenon suggests that our method avoids overfitting and holds the potential for further performance improvements with additional training data. Moreover, it's worth noting that our method doesn't necessitate GT labels for ground images during training, simplifying the process of large-scale data collection and reducing associated costs.

3 Semi-supervised Setting

Our method can be easily adapted to address the scenario when a small amount of training data with accurate pose labels is available by adding additional supervision to the network with this amount of data, *e.g.*, using the training objective in Shi *et al.* [28].

In Tab. C, we show the performance of our method when fine-tuned with different portions of the current training dataset with accurate pose labels. Data amount = 0 indicates the original weakly supervised pipeline where no such data is available. The results show that fine-tuning leads to better performance when the data amount with accurate pose labels is over 50%. However, when the data amount is smaller, the fine-tuning leads to inferior performance. We suspect this is because the model overfits the limited training data with accurate labels, causing a loss of general inference ability on other data.

A properly designed training method should address this problem. However, apart from this semi-supervised scenario, we believe the original weakly-supervised application addressed in this paper is also of high importance. It

Table C: The percentage of images whose lateral translation has been restricted to 1 meter ($d = 1$) to its ground truth value under the semi-supervised setting.

Data Amount (%)	0	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Lateral ($d = 1$)	62.73	39.49	44.61	55.31	60.96	65.60	71.48	74.61	78.61	83.86

avoids additional effort to select the portion of accurate data, as most localization algorithms and noisy GPS sensors themselves do not provide a measure of whether its prediction is accurate or not. Thus, we leave this semi-supervised setting as a future work.

4 Sharing Feature Extractors or Not at Different Scenarios

We empirically found that the feature extractors for satellite and ground images captured by a pin-hole camera are shareable in the localization task. Tab. D presents the comparison of our method with shared or non-shared feature descriptors. In this comparison, the rotation estimator is kept the same and trained only on satellite images. From the results, it can be seen that sharing weights between ground and satellite images achieves better performance. A potential explanation might be that both the satellite images and ground images captured by a pin-hole camera map straight lines in the real world to straight lines in images and the viewpoint differences of the two view images are solved by a geometry projection module. This is similar to the task of multi-view stereo and image-based rendering, where the feature extractors for multi-view images are shared, and their differences are handled by Homography/geometry warping. Not sharing weights between the two branches increases the learning burden of the network, especially when supervision is not strong, resulting in inferior performance.

While for rotation and translation estimation, we found different feature extractors for different purposes achieve better performance. Tab. E illustrates the comparison results. This might be because good features for rotation and translation estimation are not identical. When re-using the feature extractors in rotation estimation for translation, we found the network converges slowly, and the performance on both rotation and translation estimation is poor, although better than the original coarse poses that we aim to refine.

5 Performance on the Ford Dataset

We present the performance of our method on the Ford dataset in Tab. F. The results show our method achieves competitive performance with fully supervised SOTA, and sharing feature extractors between the ground and satellite branches performs better than not sharing, which is consistent with our observations on KITTI.

Table D: Sharing feature extractors or not between satellite and ground images captured by a pin-hole camera.

	Share?	Test-1 (Same-area)						Test-2 (Cross-area)					
		Lateral		Longitudinal		Azimuth		Lateral		Longitudinal		Azimuth	
		$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$
Ours	No	48.64	77.37	9.97	25.63	99.66	100.00	54.49	79.75	8.96	26.54	99.99	100.00
($\lambda=0$)	**Yes**	59.58	85.74	11.37	31.94	99.66	100.00	62.73	86.53	9.98	29.67	99.99	100.00
Ours	No	62.81	93.00	19.90	55.53	99.66	100.00	62.81	84.77	13.14	36.89	99.99	100.00
($\lambda=1$)	**Yes**	66.07	94.22	16.51	49.96	99.66	100.00	64.74	86.18	11.81	34.77	99.99	100.00

Table E: Sharing feature extractors or not for rotation and translation estimation.

	Share?	Test-1 (Same-area)						Test-2 (Cross-area)					
		Lateral		Longitudinal		Azimuth		Lateral		Longitudinal		Azimuth	
		$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$
Ours	Yes	33.77	74.66	9.17	26.13	11.69	34.64	32.18	71.92	7.58	24.00	12.64	37.24
($\lambda=0$)	**No**	59.58	85.74	11.37	31.94	99.66	100.00	62.73	86.53	9.98	29.67	99.99	100.00
Ours	Yes	35.62	81.69	10.68	31.78	10.20	31.35	32.62	73.72	8.87	26.07	10.21	31.62
($\lambda=1$)	**No**	66.07	94.22	16.51	49.96	99.66	100.00	64.74	86.18	11.81	34.77	99.99	100.00

Table F: Average results comparison on Log1 and Log2 of the Ford Multi-AV dataset.

Algorithms		Lateral		Longitudinal		Azimuth	
		$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$
Shi and Li [26]*		37.29	69.33	5.22	15.94	16.87	44.14
Shi *et al.* [28]*		63.77	82.53	**19.45**	34.17	**61.49**	92.62
Song *et al.* [33]*		46.13	-	11.97	-	58.64	-
Ours ($\lambda=0$)	share	65.47	93.50	8.00	20.11	46.63	**96.86**
	not share	14.21	69.62	5.13	13.40	46.63	**96.86**
Ours ($\lambda=1$)	share	**78.87**	**97.96**	16.42	**37.26**	46.63	**96.86**
	not share	58.19	96.83	10.14	25.33	46.63	**96.86**

6 Visual Explanation of the Spatial Correlation Process

The spatial correlation process is illiustrated in Fig. B. We first center crop the synthesized overhead view feature map depicted in Fig. 5 (b) of the main paper and make its coverage around 40×40 m^2, as we consider scene contents within 20m to the camera location is the most important for the localization purpose. Then, we adopt the cropped overhead view feature map as the correlation kernel, similar to the yellow kernel in Fig. B, and the reference satellite image as the correlation input, indicated by the green grid in Fig. B, and apply inner product between the input and the kernal. The output, indicated by the pink grid in Fig. B, is the location probability map of the ground camera with respect to the satellite image.

In practice, the coverage of the reference satellite map and the kernel is engineered to make the coverage of the convolution output slightly larger than the location search space of the ground camera. In this example, the coverage of the satellite map is around 100×100 m^2, and that of the convolution output (location probability map) is about 60×60 m^2.

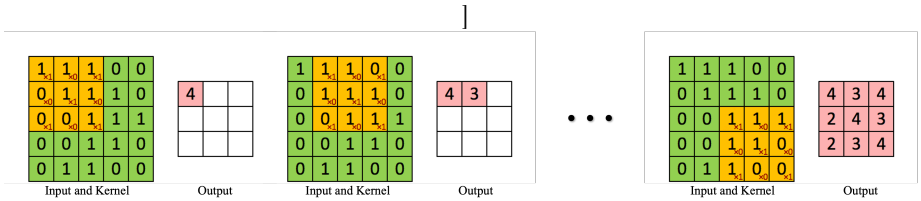


Fig. B: The spatial correlation process. We compute the inner product between the reference satellite feature map (input) and synthesized overhead view feature map (kernel) from the ground image across all possible locations. This figure is from `https://giphy.com/gifs/blog-daniel-keypoints-i4NjAwytgIRDW`

Table G: Comparison between projecting features (Feat.) and images (Imgs).

		Test-1 (Same-area)						Test-2 (Cross-area)					
		Lateral		Longitudinal		Azimuth		Lateral		Longitudinal		Azimuth	
		$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$d=1\uparrow$	$d=3\uparrow$	$\theta=1\uparrow$	$\theta=3\uparrow$
Ours ($\lambda=0$)	Imgs	45.40	80.57	6.97	21.39	99.92	100.00	48.24	79.16	6.95	20.60	100.00	100.00
	Feat	59.58	85.74	11.37	31.94	99.66	100.00	62.73	86.53	9.98	29.67	99.99	100.00
Ours ($\lambda=1$)	Imgs	54.31	90.59	13.36	38.32	99.92	100.00	58.53	87.34	11.24	33.23	100.00	100.00
	Feat.	66.07	94.22	16.51	49.96	99.66	100.00	64.74	86.18	11.81	34.77	99.99	100.00

7 Stage-wise Training VS. End-to-end Training

We present the comparison between end-to-end and stage-wise training in Tab. H. We observe that, at the beginning of end-to-end training, the rotation estimation performance for the query image is poor (near random), which negatively impacts the translation estimation performance. At the same time, allowing the signal of translation estimation loss to propagate back to the rotation estimator negatively affects its performance. Furthermore, end-to-end training requires large GPU memory and, thus, a smaller batch size, which is detrimental to metric learning performance.

Table H: Comparison with end-to-end training on KITTI dataset (Ours with $\lambda=0$)

Training Approach	Test-1 (Same-area)			Test-2 (Cross-area)		
	Lateral $d=1$	Longitudinal $d=1$	Azimuth $\theta=1^\circ$	Lateral $d=1$	Longitudinal $d=1$	Azimuth $\theta=1^\circ$
End-to-end	32.83	8.22	10.15	32.53	7.76	10.10
Stage-wise	**59.58**	**11.37**	**99.66**	**62.73**	**9.98**	**99.99**

8 Angle ambiguity at 0°& 360°, why not rotation matrix?

Since we have prior knowledge of the coarse orientation, the angle ambiguity can be restricted to be smaller than 360°. Thus, the angle ambiguity at 0° and 360° can be avoided. A rotation matrix over-parameterizes the 1-DoF rotation

angle in the cross-view image matching, thus leading to inferior performance, as shown in the 1st row of Tab. I. We further make the pose regressor output sine and cosine of the angle (2-DoF output). The results in the 2nd row of Tab. I are promising but still inferior to our original parameterization (last row).

Table I: Additional ablation study on rotation estimation on the KITTI dataset.

Rotation Parameterization	Test-1 (Same-area)		Test-2 (Cross-area)	
	$\theta = 1°$	$\theta = 3°$	$\theta = 1°$	$\theta = 3°$
Rotation Matrix	5.05	15.44	5.07	15.51
Sin(angle) & Cos(angle)	90.01	**100.00**	86.52	**100.00**
Angle	**99.66**	**100.00**	**99.99**	**100.00**

9 Comparison Between Projecting Features and Images

In this paper, we follow the general practice of projecting features instead of images [32]. This is because when projecting ground images to an overhead view by assuming ground plane Homography, the pixels for scene objects above the ground plane are incorrectly projected to the overhead view and thus suffer distortion. In this way, the scene information of these objects will be lost in the projected image, resulting in inferior localization performance.

In contrast, features have a larger field of view of the original image and encode higher-level semantic information about the scene. For example, the building roots also have a semantic meaning of "building". It can be mapped to the building roof in the overhead view, which shares the same semantic information as the building root. Thus, projecting features instead of the original images can tolerate the errors in the overhead-view projection by the ground plane Homography to some extent. We illustrate the experimental comparison between projecting features and images in Tab. G. Not surprisingly, projecting features achieves better performance.

10 Model Size and Evaluation Speed Comparison

We present the model size and evaluation speed comparison with two recent state-of-the-art, whose models and evaluation scripts have been released, in Tab. J. All of them are evaluated on an RTX 3090 GPU. It can be seen that our method achieves the fastest evaluation speed with a relatively small model size.

11 Limitations

Although our self-supervised learning approach has achieved promising results, it has a few limitations.

(i) First, as explained previously, our self-supervised training strategy for rotation estimation is only suitable for ground images captured by a pin-hole

Table J: Model size and evaluation speed comparison on the KITTI dataset.

Model Size			Evaluation Speed		
Shi and Li [26]	Shi *et al.* [28]	**Ours**	Shi and Li [26]	Shi at al. [28]	**Ours**
20.2 M	29.1 M	20.6 M	500 ms	200 ms	47 ms

camera. Due to the significant domain differences between panoramic and satellite images, it cannot be applied to estimating a spherical camera's orientation.

(ii) Second, our deep metric learning supervision strategy computes the spatial correlation between each query image and several satellite images. To save GPU memory and enable a reasonable batch size for metric learning, we use the feature level of a quarter of the original image size for the translation estimation. This actually sacrifices localization accuracy to some extent.

(iii) For the same reason, we cannot adopt a more sophisticated overhead-view feature synthesis method because it will consume significant memory, thus sacrificing batch size, and the weak supervision limits the learning ability of a powerful overhead-view synthesis module with complex designs.

(vi) Finally, similar to all the ground-to-satellite localization networks where a single camera is used for query, our method suffers poor localization performance along the longitudinal direction, as shown in Fig. C. This can potentially be addressed using a video or multi-camera setup for the query.

We leave these unsolved problems as our future work and encourage the community to pay attention to them.

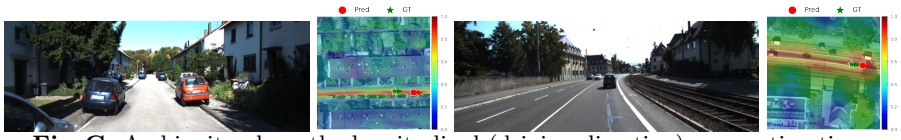


Fig. C: Ambiguity along the longitudinal (driving direction) pose estimation.

12 Potential Negative Impact

This paper introduces a novel approach for self-supervised ground-to-satellite image registration. The objective of this approach is to enhance the accuracy of coarse camera pose estimates, such as those obtained from noisy GPS sensors, through ground-to-satellite image matching. However, it also raises concerns about privacy, particularly regarding the potential for individuals or sensitive locations to be identified and tracked without their consent. Unauthorized access to satellite imagery could enable surveillance activities or intrusions into personal privacy, raising ethical and legal implications.

We emphasize that the proposed method should be utilized in a manner that aligns with legal and ethical considerations. Careful implementation and

adherence to privacy regulations and policies are crucial to ensure the ethical and responsible use of this approach.