

PIDLoc: Cross-View Pose Optimization Network Inspired by PID Controllers

Supplementary Material

A. Implementation details

We provide additional implementation details in Section 4 (Experiments). The resolutions of the ground-view images are 375×1242 and 432×816 for the KITTI and FMAVS datasets, respectively. The experiments were conducted in the HighlyAccurate [19] and SIBCL [30] settings for the KITTI and FMAVS datasets, respectively.

Both the HighlyAccurate and SIBCL settings generate initial poses by adding noise to the ground-truth pose. In the HighlyAccurate setting, the initial pose is aligned to the satellite image center. In contrast, the SIBCL setting aligns the ground-truth pose to the satellite image center, which risks overfitting the model by biasing predictions toward the center. To ensure a fair comparison on the FMAVS dataset, we modified the SIBCL setting by not aligning the ground-truth pose to the center and reproduced previous works.

In the I branch, the search radius and step size of pose candidates are set to one-fourth of the noise range of the initial pose. This configuration generates two samples along the lateral, longitudinal, and azimuth directions; resulting in a total of six pose candidates. For example, if the noise range of the initial pose is $\pm 20\text{m}$, $\pm 20\text{m}$, and $\pm 10^\circ$, the search radius and step size are set to 10m , 10m , and 5° , respectively. Excluding the given pose, two samples are generated in each direction: $+10\text{m}$ and -10m in the position direction, and $+5^\circ$ and -5° in the orientation direction. To prevent gradient explosion during training, the gradients of the network parameters are clipped to a maximum ℓ_2 -norm of 10.

B. Additional experiments

B.1. Ablation studies

Impact of the coefficients of the PID branches We evaluated the impact of the coefficients k_p , k_i , and k_d of the PID branches on the localization performance using the cross-view KITTI dataset. In the conventional control theory, PID controllers are highly sensitive to the coefficients, requiring manual gain tuning to optimize the performance. In contrast, the proposed PIDLoc optimizes these coefficients through a learning-based approach, eliminating the need for manual hyperparameter tuning.

Table F compares the performance of constant and learnable coefficients in the PID branches. The constant coefficients (k_p , k_i , k_d) are set to one and the learnable coefficients (k_p , k_i , k_d) are initialized to one. After the training, the learnable coefficients were tuned to 0.868, 0.930, and 1.214, respectively. This learning-based tuning resulted

Learnable	Lat. (%) \uparrow R@1m	Long. (%) \uparrow R@1m	Orien. (%) \uparrow R@1°
X	69.51	48.94	99.95
✓	71.01	50.02	99.96

Table A. Comparison between constant and learnable coefficients in the PID branches.

Branches	k_p	k_i	k_d	Lat. (%) \uparrow R@1m	Long. (%) \uparrow R@1m	Orien. (%) \uparrow R@1°
P	1.0	0.0	0.0	66.67	41.03	99.93
PI	1.0	1.0	0.0	68.36	48.13	99.81
PD	1.0	0.0	1.0	67.92	46.78	99.96
PID	1.0	1.0	1.0	69.51	48.94	99.95
PID	0.5	1.0	1.0	69.94	49.82	99.88
PID	1.5	1.0	1.0	68.41	48.40	99.95
PID	1.0	0.5	1.0	68.36	48.51	99.94
PID	1.0	1.5	1.0	68.45	49.41	99.95
PID	1.0	1.0	0.5	69.07	48.21	99.97
PID	1.0	1.0	1.5	68.40	49.93	99.89

Table B. Ablation analysis of the constant coefficients in the PID branches.

in a lateral and longitudinal localization improvement of 1.50%p and 1.08%p compared with the constant coefficient baseline. The learnable coefficients adaptively capture the importance of each branch, improving the overall localization performance.

Table B compares the impact of the coefficients on localization performance by fixing their values to some constants. The PID branches consistently improve position performance compared with the P, PI, and PD branches, demonstrating the effectiveness of including ID branches across various coefficient settings. Notably, configurations of ($k_p = 0.5$, $k_i = 1.0$, $k_d = 1.0$) and ($k_p = 1.0$, $k_i = 1.0$, $k_d = 1.5$) indicate that reducing k_p improves lateral performance while increasing k_d improves longitudinal performance. This result is consistent with the learnable coefficients of ($k_p = 0.868$, $k_i = 0.930$, $k_d = 1.214$) in Table F, demonstrating that the learnable coefficients effectively capture the balance among the PID branches.

Impact of pose candidates In Table C, we evaluated the impact of the number of pose candidates in the I branch under two initial pose error settings: $\pm 20\text{m}$ and $\pm 10^\circ$, and $\pm 30\text{m}$ and $\pm 15^\circ$. The number of pose candidates is set to zero, two, and four per direction. The case with zero candi-

Pose candidates per direction	Pose noise	Lat. (%) \uparrow R@1m	Long. (%) \uparrow R@1m	Orien. (%) \uparrow R@1°
0	$\pm 20\text{m}, \pm 10^\circ$	67.90	46.80	99.91
2	$\pm 20\text{m}, \pm 10^\circ$	71.01	50.02	99.96
4	$\pm 20\text{m}, \pm 10^\circ$	69.21	51.24	99.91
0	$\pm 30\text{m}, \pm 15^\circ$	60.60	29.37	97.46
2	$\pm 30\text{m}, \pm 15^\circ$	62.12	37.08	97.54
4	$\pm 30\text{m}, \pm 15^\circ$	62.24	38.94	98.46

Table C. Ablation analysis of the number of the pose candidates in the I branch.

Iterations	Lat. (%) \uparrow R@1m	Long. (%) \uparrow R@1m	Orien. (%) \uparrow R@1°	Inference time (ms)
1	56.48	32.05	99.92	101
3	70.13	47.68	99.95	225
5	71.01	50.02	99.96	374
7	72.45	51.24	99.62	473

Table D. Impact of the iterations of the PIDLoc on the localization performance.

date corresponds to the PD branches.

Table C demonstrates that increasing the number of pose candidates significantly enhances localization performance. Specifically, under the $\pm 30\text{m}$ initial noise conditions, incorporating two and four pose candidates per direction improves longitudinal performance by 7.71%p and 9.57%p, respectively, compared with the case of no pose candidates. The process of sampling pose candidates adds approximately 22ms of inference time and 0.15GB of GPU memory per candidate. These results indicate that incorporating more pose candidates enables the model to integrate global context effectively, but a balance is required to manage computational resources.

Impact of iterations In Table D, we evaluated the impact of the iterations on localization performance using the cross-view KITTI dataset. The proposed PIDLoc iteratively refines the estimated pose toward the ground-truth pose by leveraging the cross-view features at the given pose. When increasing iterations from one to three, the recall rates were improved by 13.65%p and 15.63%p in the lateral and longitudinal directions, respectively. However, Table D shows that performance gains diminish beyond five iterations, as the refinement process converges and additional iterations rarely provide new information for further pose adjustments. To balance the performance with computational efficiency, we adopt the five iterations as the default configuration.

Branch	Parallel candidates	Inference time (ms)	Memory (GB)
P branch	-	182	5.61
PD branches	-	251	6.41
PI branches	X	370	6.53
PI branches	\checkmark	234	7.13
PID branches	X	510	7.26
PID branches	\checkmark	374	7.32

Table E. Computational complexity analysis of the proposed PID-Loc.

B.2. Computational resources

We analyzed the computational complexity of the PID-Loc across different PID-branch configurations. The computational complexity was measured using inference time per image and GPU memory usage of the PIDLoc on an NVIDIA RTX A5000 GPU and an AMD EPYC 7453 28-Core Processor CPU. When performing iterative sampling of pose candidates, the inference time for the PID branches is 510ms per image, which is comparable to the 500ms reported for HighlyAccurate [19]. When performing GPU parallel processing for pose candidate sampling, the inference time for the PID branches is reduced to 374ms per image, achieving 26.67% reduced inference time with only 0.83% increase in GPU memory usage. Notably, both iterative and parallel sampling methods use the same approach for selecting pose candidates, ensuring consistent recall rates and localization performance. These results demonstrate the computational efficiency of the PIDLoc, enabling real-world applications such as autonomous navigation and robotics.

C. Qualitative analysis

C.1. Pose update on the diverse initial poses

Figure A illustrates the position adjustment of the PID branches at a given pose. The P branch only focuses on the given pose, resulting in the convergence to a local optimum. Specifically, the P branch often predicts the position in the wrong direction or fails to update the pose when trapped in a local optimum. In contrast, the I branch incorporates the wider field of view (FoV) from the pose candidates, making them converge to the global optimum. The I branch successfully updates the position toward the ground-truth pose for most initial poses. However, it struggles to accurately update poses in regions near the ground-truth pose. The D branch utilizes feature difference gradients to update the position with high precision. The feature difference gradients capture subtle feature variations near the ground-truth pose, enabling accurate pose updates.

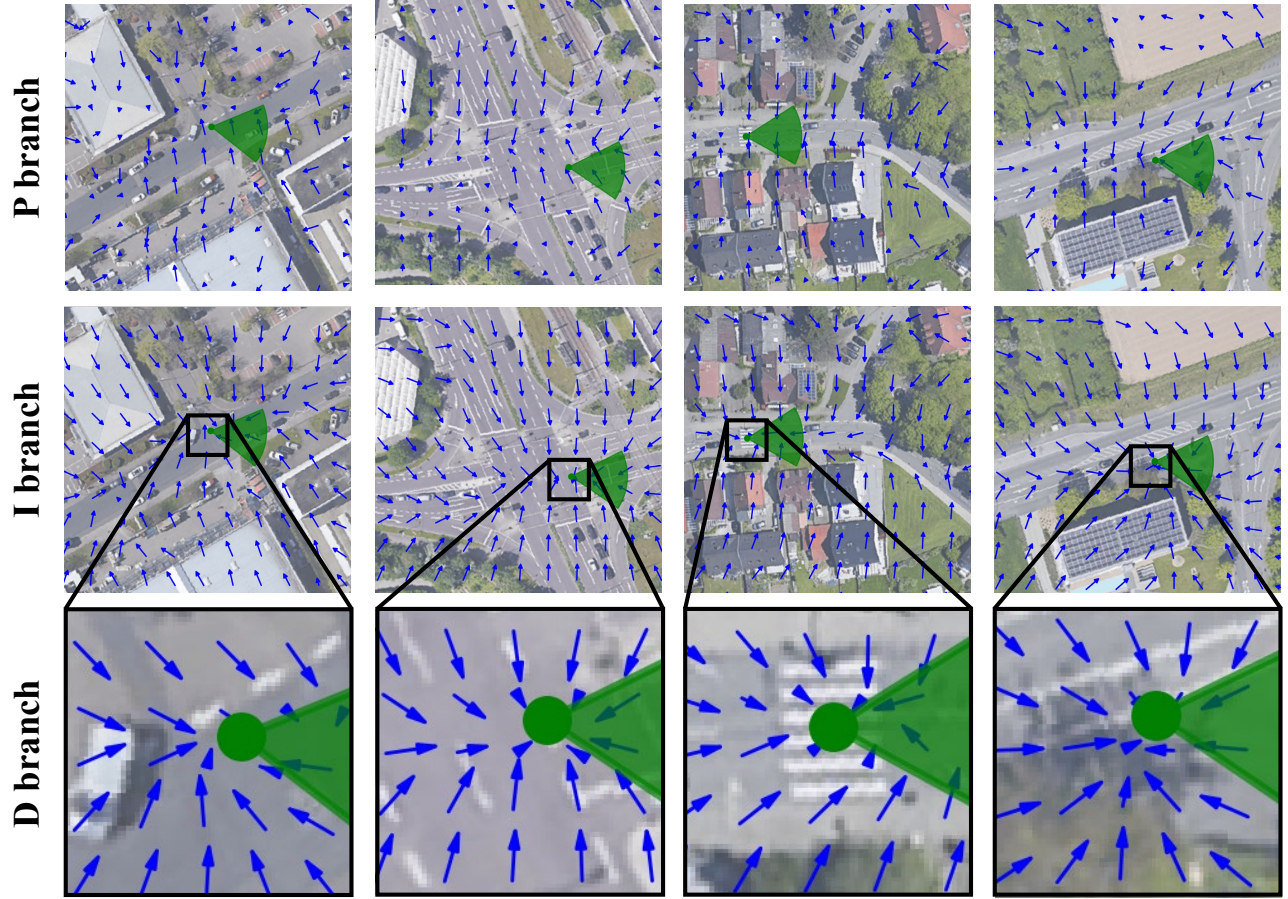


Figure A. The PIDLoc performs localization by incorporating local, global, and fine-grained contexts. The green circular sector represents the ground truth pose. The blue arrow represents the position adjustment of the given pose during the single iteration. The length of the blue arrow indicates the position update size. Similar to existing methods, the P branch relies solely on the given pose, often resulting in the convergence to a local optimum. The I branch incorporates global context from diverse poses, enabling position updates toward the ground-truth pose across a wide range of initial poses. The D branch leverages gradients of feature differences to perform fine-grained pose adjustments.

C.2. Visualization of localization results

In Figure B, we compared the SIBCL [30] with our PIDLoc under an initial pose error of $\pm 30\text{m}$ and $\pm 15^\circ$ on the KITTI dataset. Figures B (a)-(d) illustrate examples with repetitive patterns along the lateral and longitudinal directions. In case (a), where the initial lateral pose error is significant, the SIBCL only finds a local optimum. In contrast, the PIDLoc accurately finds the global optimum by avoiding buildings that resemble roads. Cases (b)-(d) show that the SIBCL accurately estimates the lateral direction but the estimation converges to a local optimum in the longitudinal direction. This limitation arises from its restricted FoV when dealing with repetitive patterns along the longitudinal direction. In contrast, the PIDLoc leverages global and fine-grained contexts, enabling accurate localization even in

Positional Embedding	Lat. (%) \uparrow R@1m	Long. (%) \uparrow R@1m	Orien. (%) \uparrow R@1 $^\circ$
X	66.06	42.37	99.88
\checkmark	71.01	50.02	99.96

Table F. Comparison of SPE with and without positional embedding.

challenging scenarios with repetitive patterns, such as building facades and vegetation rows.

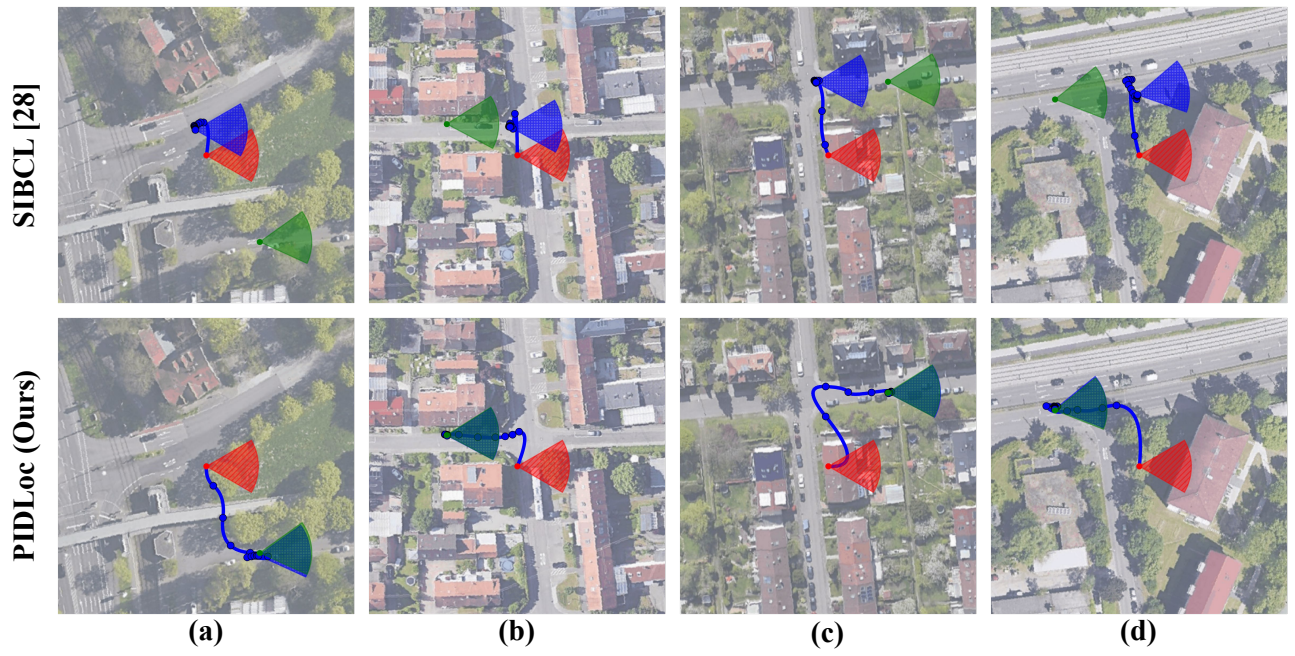


Figure B. Visualization of localization results. The red, green, and blue circular sectors represent the current, ground-truth, and predicted pose, respectively. The blue line represents the iterative trajectory of predicted poses and the blue dot represents the predicted pose at each iteration. Compared with SIBCL [30], PIDLoc more accurately finds the global optimum in a challenging environment with repetitive patterns.