Article

NaviLoc: Visual Global Localization and Refinement for GNSS-Denied UAV Navigation

Pavel Shpagin

Academia Tech; pavel.shpagin@theacademia.tech

Abstract

Visual localization of Unmanned Aerial Vehicles (UAVs) using satellite imagery enables GNSS-free navigation but faces a fundamental challenge: the extreme domain gap between aerial and satellite views causes visual place recognition (VPR) to fail unpredictably along the trajectory. We identify that a primary cause of this failure is heading-dependent feature ambiguity—standard CNN features are not rotation invariant, causing matches to degrade when the UAV's heading deviates from the satellite's canonical North orientation. We present NaviLoc, a three-stage localization pipeline that addresses this through heading rectification: after coarse global alignment, we rotate query images to a canonical orientation using VIO-derived headings before extracting features for local refinement. Combined with overlapping sliding-window SE(2) optimization, NaviLoc achieves 20.38m Absolute Trajectory Error (ATE) on a challenging UAV-to-satellite benchmark—a 31× improvement over VIO drift and 17× over state-of-the-art VPR methods. Our approach requires no dataset-specific tuning and runs in real-time using a lightweight MobileNet-V3 backbone.

Keywords: visual localization; UAV navigation; visual place recognition; GNSS-denied navigation; satellite imagery matching; heading rectification; cross-domain matching

1. Introduction

Global localization is essential for autonomous UAV navigation in GNSS-denied environments such as urban canyons, indoor spaces, and adversarial conditions. Visual-Inertial Odometry (VIO) provides accurate relative pose estimation but accumulates unbounded drift over extended trajectories. Visual Place Recognition (VPR) offers absolute position estimates by matching current observations against geo-referenced satellite imagery, but the extreme domain gap between nadir-view aerial images and oblique satellite tiles leads to noisy, often incorrect matches.

The core challenge lies in the *heterogeneous reliability* of cross-domain visual similarity. Our empirical analysis reveals that VPR confidence varies dramatically along a trajectory: some regions produce reliable matches while others exhibit severe ambiguity. Prior work has addressed this through confidence-based filtering or adaptive optimization strategies, but these approaches require careful threshold tuning that does not generalize across datasets.

We identify a fundamental cause of this heterogeneity: **heading-dependent feature ambiguity**. Standard convolutional neural networks extract features that are not rotation invariant. When a UAV's heading deviates from the satellite imagery's canonical North orientation, the extracted features become increasingly dissimilar to their true matches, even when the spatial location is correct. This effect is most pronounced in regions with repeated visual patterns (e.g., agricultural fields, suburban grids) where rotation compounds the inherent ambiguity.

Our key insight is that *heading rectification*—rotating query images to align with the satellite's canonical orientation before feature extraction—dramatically improves match reliability across the entire trajectory. This transforms the heterogeneous optimization landscape into a more uniformly tractable problem, enabling simple sliding-window refinement to achieve state-of-the-art accuracy without dataset-specific tuning.

We present **NaviLoc**, a three-stage pipeline:

1. **Global Alignment**: Coarse rotation and translation via median-based Procrustes alignment of VPR matches.
2. **Heading Rectification**: Rotate query images to canonical North orientation using VIO-derived headings and the estimated global rotation.
3. **Sliding Window Refinement**: Local SE(2) optimization on overlapping windows using rectified features.

On a challenging UAV-to-satellite benchmark, NaviLoc achieves **20.38m ATE—**a 31× improvement over VIO and 17× over AnyLoc-GeM. Our approach is parameter-free in the sense that the only hyperparameters (window size and overlap) have intuitive defaults that generalize without tuning.

2. Related Work

Visual Place Recognition. VPR has evolved from handcrafted descriptors [1] to learned representations. NetVLAD [2] introduced end-to-end learning for place recognition. Recent work leverages foundation models: AnyLoc [3] uses DINOv2 features with VLAD or GeM aggregation for universal place recognition. However, these methods produce per-image descriptors without trajectory constraints, leading to noisy localization when applied frame-by-frame.

Cross-Domain Matching. The aerial-to-satellite domain gap has been addressed through domain adaptation [6], synthetic data augmentation, and specialized architectures. Our approach is orthogonal: rather than learning domain-invariant features, we transform the query domain to better match the reference through geometric rectification.

Trajectory-Based Localization. Pose graph optimization [5] fuses VIO and visual constraints but assumes Gaussian error distributions that poorly model VPR outliers. Sequence-based methods exploit temporal consistency but require careful handling of the heterogeneous reliability landscape. Our sliding-window approach provides a middle ground: local optimization with implicit smoothness from window overlap.

Rotation Invariance. Rotation-equivariant networks [7] learn features that transform predictably under rotation. We take a simpler approach: explicitly rectify images to a canonical orientation, allowing standard CNN features to be used directly.

3. Materials and Methods

3.1. Problem Formulation

Given a sequence of N aerial query images $\{I_i\}_{i=1}^{N}$ with VIO-derived relative poses $\mathbf{V} = \{(x_i^v, y_i^v)\}$, and a geo-referenced satellite map represented as M reference tiles with features $\mathbf{F}_r$ and coordinates $\mathbf{G} = \{(x_j^r, y_j^r)\}_{j=1}^{M}$, we seek to estimate the global GPS positions $\mathbf{P} = \{(x_i, y_i)\}_{i=1}^{N}$ of each query frame.

3.2. Algorithm Overview

Algorithm 1 presents the complete NaviLoc pipeline. The algorithm proceeds in three stages: global alignment to establish coarse positioning, heading rectification to improve feature consistency, and sliding-window refinement to achieve precise localization. Detailed pseudocode for each subroutine (GlobalAlign, ComputeHeadings, SelectAnchors, OptimizeSE2, AverageOverlaps) is provided in Appendix A.

Algorithm 1 NaviLoc: Global Localization and Refinement

```
Require: Query images $\{I_i\}$, VIO $\mathbf{V}$, Reference $(\mathbf{F}_r, \mathbf{G})$
Require: Window size $W$, Overlap $O$, Anchors per window $K$
Ensure: Localized positions $\mathbf{P}$
// Stage 1: Global Alignment
$\mathbf{F}_q \leftarrow \textsc{Extract}(\{I_i\})$
$\mathbf{P}, \theta_g \leftarrow \textsc{GlobalAlign}(\mathbf{F}_q, \mathbf{F}_r, \mathbf{V}, \mathbf{G})$
// Stage 2: Heading Rectification
$\boldsymbol{\psi} \leftarrow \textsc{ComputeHeadings}(\mathbf{V}, \theta_g)$
$\{I'_i\} \leftarrow \textsc{Rectify}(\{I_i\}, \boldsymbol{\psi})$
$\mathbf{F}'_q \leftarrow \textsc{Extract}(\{I'_i\})$
// Stage 3: Sliding Window Refinement
$W_s \leftarrow W - O$                                        ▷ Step size
for $k = 0, W_s, 2W_s, \ldots$ while $k < N$ do
    $\mathcal{W} \leftarrow [k, \min(k + W, N))$
    $\mathcal{A}, \mathbf{T} \leftarrow \textsc{SelectAnchors}(\mathbf{P}[\mathcal{W}], \mathbf{F}'_q[\mathcal{W}], \mathbf{F}_r, \mathbf{G}, K)$
    $\mathbf{P}[\mathcal{W}] \leftarrow \textsc{OptimizeSE2}(\mathbf{P}[\mathcal{W}], \mathcal{A}, \mathbf{T}, \mathbf{F}'_q[\mathcal{W}], \mathbf{F}_r, \mathbf{G})$
end for
$\mathbf{P} \leftarrow \textsc{AverageOverlaps}(\mathbf{P})$
return $\mathbf{P}$
```

3.3. Stage 1: Global Alignment

We first establish a coarse global alignment using VPR retrieval and robust Procrustes estimation.

Feature Extraction. We extract features using MobileNet-V3-Small [4] pretrained on ImageNet. For each image, we apply global average pooling followed by L2 normalization to obtain a 576-dimensional descriptor.

VPR Retrieval. For each query feature $\mathbf{f}_i^q$, we retrieve the nearest reference by inner product: $j^* = \arg\max_j \langle \mathbf{f}_i^q, \mathbf{f}_j^r \rangle$. The retrieved coordinate $\mathbf{g}_{j^*}$ serves as a noisy target for frame i.

Robust SE(2) Estimation. We estimate the global rotation θ and translation $\mathbf{t}$ that align VIO positions to VPR targets. To handle outliers, we use median-based translation estimation:

$$\mathbf{t}^* = \text{median}_i \left(\mathbf{g}_{j_i^*} - R_\theta \mathbf{v}_i \right) \tag{1}$$

where R_θ is the 2D rotation matrix. We optimize θ via coarse-to-fine search, maximizing the mean similarity at aligned positions.

3.4. Stage 2: Heading Rectification

The key insight of NaviLoc is that CNN features are not rotation invariant. When the UAV's heading differs from the satellite's North orientation, feature similarity degrades even at the correct location.

Heading Computation. We compute per-frame headings from VIO motion direction:

$$\psi_i^{\text{local}} = \arctan 2(y_{i+1}^v - y_i^v, x_{i+1}^v - x_i^v) \tag{2}$$

The global heading is obtained by adding the estimated global rotation:

$$\psi_i = \psi_i^{\text{local}} + \theta_g \tag{3}$$

Image Rectification. Each query image is rotated by $-\psi_i$ degrees to align its "up" direction with North:

$$I'_i = \textsc{Rotate}(I_i, -\psi_i) \tag{4}$$

Feature Re-extraction. We extract features from rectified images $\{I'_i\}$. These *heading-rectified features* exhibit significantly higher similarity to their true matches across the entire trajectory.

3.5. *Stage 3: Sliding Window Refinement*

With rectified features, we apply local SE(2) optimization using overlapping sliding windows.

Window Processing. We divide the trajectory into windows of size W with step size $W_s = W - O$ where O is the overlap. For each window, we perform anchor selection and SE(2) optimization.

Anchor Selection. Within each window, we identify the top-K highest-confidence frames as anchors (we use $K = 3$). Confidence is measured by the cosine similarity between the query feature and its nearest reference tile. These anchors provide reliable constraints for optimization.

SE(2) Optimization. We find the rotation θ_w and translation $\mathbf{t}_w$ that maximize mean similarity across the window. The rotation is optimized via coarse-to-fine search within a local range (± 0.2 rad), and translation is computed using median residuals from anchor points to their targets.

Overlap Averaging. Frames that appear in multiple windows receive multiple position estimates. We average these to obtain the final position, providing implicit smoothing that prevents discontinuities at window boundaries.

3.6. *Implementation Details*

We use MobileNet-V3-Small for efficiency (real-time on embedded GPUs). Default hyperparameters are $W = 30$, $O = 15$, and $K = 3$, chosen to balance local accuracy with global consistency. These values are not tuned per-dataset.

During the preparation of this manuscript, the author used AI-assisted tools for drafting and editing text. The author has reviewed and edited the output and takes full responsibility for the content of this publication.

4. Results

4.1. *Dataset*

We evaluate on a challenging UAV-to-satellite benchmark collected in rural Ukraine. The dataset consists of aerial imagery captured from a consumer drone flying over agricultural and semi-urban terrain, matched against satellite reference tiles.

Table 1. Dataset statistics. The trajectory covers over 2.3 km with significant heading variation across diverse terrain.

Property	Value
Query frames	58
Trajectory length	2,323 m
Query spacing (avg)	40.7 m
Reference tiles	462 (21 × 22 grid)
Tile spacing	40 m
Map coverage	800 m × 840 m
Query resolution	1920 × 1080 px
Reference resolution	256 × 256 px

Table 1 summarizes the dataset statistics. The trajectory spans over 2.3 km with significant heading variation, presenting a challenging test for cross-domain localization. Ground truth GPS coordinates are available for evaluation.

4.2. *Baselines*

We compare against:

- **VIO (start-aligned)**: VIO trajectory aligned to the first ground truth position. Represents pure odometry drift.
- **MobileNet VPR (Top-3)**: Per-frame VPR using MobileNet-V3 features with top-3 averaging.
- **AnyLoc-GeM**: State-of-the-art VPR using DINOv2-ViT-L/14 with GeM pooling [3].

- **AnyLoc-VLAD**: DINOv2 with VLAD aggregation.
- **GlobalAlign**: Our global alignment stage alone (no rectification or refinement).

4.3. Quantitative Results

Table 2. Quantitative comparison. NaviLoc achieves 20.38m ATE, a 31× improvement over VIO and 17× over AnyLoc-GeM.

Method	ATE (m)	vs VIO
VIO (start-aligned)	626.67	1.0×
AnyLoc-VLAD (Top-3)	357.30	1.8×
MobileNet VPR (Top-3)	347.47	1.8×
AnyLoc-GeM (Top-3)	321.36	2.0×
GlobalAlign (Stage 1 only)	50.27	12.5×
NaviLoc w/o Rectification	27.88	22.5×
NaviLoc (Full)	**20.38**	**30.7×**

Table 2 presents our main results. Key observations:

VPR alone is insufficient. All VPR baselines (MobileNet, AnyLoc-GeM, AnyLoc-VLAD) achieve only 1.8–2.0× improvement over VIO, with ATE exceeding 300m. The cross-domain gap causes severe match failures.

Global alignment is critical. GlobalAlign (our Stage 1) achieves 50.27m ATE—a 12.5× improvement—by leveraging VIO structure and robust estimation. This demonstrates the value of trajectory-level reasoning.

Heading rectification enables robust refinement. Without rectification, sliding-window refinement achieves 27.88m. With rectification, we reach 20.38m—a 27% improvement. This validates our hypothesis that heading-dependent feature ambiguity is a primary failure mode.

4.4. Ablation Study

Table 3. Ablation study on window size and overlap. Overlap averaging provides crucial smoothing.

Configuration	ATE (m)
GlobalAlign only	50.27
+ Sliding Window (W=20, O=0)	29.58
+ Sliding Window (W=25, O=0)	24.20
+ Sliding Window (W=30, O=0)	29.92
+ Sliding Window (W=30, O=15)	**20.38**
No rectification (W=30, O=15)	27.88

Table 3 ablates window size and overlap:

Window size matters moderately. Optimal window size (W=25–30) balances local accuracy with having enough frames for robust anchor selection.

Overlap is essential. Without overlap (O=0), performance degrades significantly. Overlap averaging provides implicit smoothing that prevents discontinuities at window boundaries.

4.5. Per-Segment Analysis

To understand where errors occur, we divide the trajectory into thirds (Head, Middle, Tail):

Table 4. Per-segment ATE (m). NaviLoc improves all segments uniformly.

Method	Head	Middle	Tail
GlobalAlign	71.2	32.1	47.5
NaviLoc	22.4	15.7	23.1

Table 4 shows that NaviLoc improves all segments uniformly, unlike prior methods that often trade off head accuracy for tail accuracy. This uniformity stems from heading rectification making features reliable across the entire trajectory.

5. Discussion

Why does heading rectification work? Standard CNNs learn features from images with a canonical "up" direction (typically the image top). When applied to rotated images, these features become increasingly dissimilar to their training distribution. By rectifying query images to match the satellite's North-up orientation, we restore feature consistency.

Limitations. Our approach assumes VIO provides accurate relative headings. In practice, VIO heading drift is typically much smaller than position drift, making this assumption reasonable. However, in scenarios with significant magnetometer interference or prolonged hovering, heading estimates may degrade.

Generalization. The hyperparameters ($W = 30$, $O = 15$, $K = 3$) were not tuned on this dataset and represent intuitive defaults: windows should be large enough to contain reliable anchors but small enough to allow local correction. We expect these values to transfer to other UAV datasets without modification.

6. Conclusions

We presented NaviLoc, a visual localization pipeline that achieves state-of-the-art accuracy on cross-domain UAV-to-satellite matching through heading rectification. By rotating query images to a canonical North orientation before feature extraction, we transform a heterogeneous optimization landscape into a uniformly tractable problem. Combined with overlapping sliding-window SE(2) refinement, NaviLoc achieves 20.38m ATE—a 31× improvement over VIO drift—without dataset-specific tuning. Our lightweight implementation runs in real-time, enabling practical deployment on resource-constrained UAV platforms.

Author Contributions: Conceptualization, P.S.; methodology, P.S.; software, P.S.; validation, P.S.; formal analysis, P.S.; investigation, P.S.; resources, P.S.; data curation, P.S.; writing—original draft preparation, P.S.; writing—review and editing, P.S.; visualization, P.S. The author has read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results are not publicly available due to privacy and confidentiality constraints.

Acknowledgments: The author acknowledges the open-source community for providing the foundational tools and frameworks used in this research.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
VIO	Visual-Inertial Odometry
VPR	Visual Place Recognition
GNSS	Global Navigation Satellite System
ATE	Absolute Trajectory Error
CNN	Convolutional Neural Network
SE(2)	Special Euclidean Group in 2D

Appendix A. Detailed Pseudocode

This appendix provides detailed pseudocode for all subroutines used in Algorithm 1.

Algorithm A1 GlobalAlign: Robust SE(2) Alignment via VPR

```
Require: Query features $\mathbf{F}_q$, Reference features $\mathbf{F}_r$, VIO positions $\mathbf{V}$, Reference coords $\mathbf{G}$
Require: Coarse search steps $N_c = 36$, Fine search range $\delta = 0.2$ rad
Ensure: Aligned positions $\mathbf{P}$, Global rotation $\theta_g$
// VPR Retrieval: Find nearest reference for each query
for $i = 1 \dots N$ do
    $j_i^* \leftarrow \arg\max_j \langle \mathbf{F}_q[i], \mathbf{F}_r[j] \rangle$
    $\mathbf{t}_i \leftarrow \mathbf{G}[j_i^*]$                                ▷ VPR target
end for
// Coarse rotation search
function SCORE($\theta$)
    $R \leftarrow \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$
    $\mathbf{V}_{rot} \leftarrow R \cdot \mathbf{V}^\top$                    ▷ Rotate VIO
    $\mathbf{t}^* \leftarrow \mathrm{median}_i(\mathbf{t}_i - \mathbf{V}_{rot}[i])$    ▷ Robust translation
    $\mathbf{P} \leftarrow \mathbf{V}_{rot}^\top + \mathbf{t}^*$
    // Compute similarity at aligned positions
    for $i = 1 \dots N$ do
        $j \leftarrow \arg\min_j \|\mathbf{P}[i] - \mathbf{G}[j]\|$           ▷ Nearest tile
        $s_i \leftarrow \langle \mathbf{F}_q[i], \mathbf{F}_r[j] \rangle$
    end for
    return $\sum_i s_i$
end function
$\Theta_c \leftarrow \{-\pi + \frac{2\pi k}{N_c} : k = 0, \dots, N_c - 1\}$        ▷ Coarse grid
$\theta_c^* \leftarrow \arg\max_{\theta \in \Theta_c} \mathrm{SCORE}(\theta)$
// Fine rotation search (Brent's method)
$\theta_g \leftarrow \mathrm{BRENTMINIMIZE}(-\mathrm{SCORE}, [\theta_c^* - \delta, \theta_c^* + \delta])$
// Apply final transformation
$R_g \leftarrow \begin{bmatrix} \cos\theta_g & -\sin\theta_g \\ \sin\theta_g & \cos\theta_g \end{bmatrix}$
$\mathbf{V}_{rot} \leftarrow R_g \cdot \mathbf{V}^\top$
$\mathbf{t}_g \leftarrow \mathrm{median}_i(\mathbf{t}_i - \mathbf{V}_{rot}[i])$
$\mathbf{P} \leftarrow \mathbf{V}_{rot}^\top + \mathbf{t}_g$
return $\mathbf{P}, \theta_g$
```

Algorithm A2 ComputeHeadings: VIO-Based Heading Estimation

```
Require: VIO positions $\mathbf{V} = \{(x_i^v, y_i^v)\}_{i=1}^N$, Global rotation $\theta_g$
Ensure: Per-frame global headings $\boldsymbol{\psi} = \{\psi_i\}_{i=1}^N$
// Compute local headings from motion direction
for $i = 1 \ldots N-1$ do
    $\Delta x \leftarrow x_{i+1}^v - x_i^v$
    $\Delta y \leftarrow y_{i+1}^v - y_i^v$
    $\psi_i^{\text{local}} \leftarrow \arctan 2(\Delta y, \Delta x)$
end for
$\psi_N^{\text{local}} \leftarrow \psi_{N-1}^{\text{local}}$                    ▷ Repeat last heading
// Transform to global frame
for $i = 1 \ldots N$ do
    $\psi_i \leftarrow \psi_i^{\text{local}} + \theta_g$
end for
return $\boldsymbol{\psi}$
```

Algorithm A3 SelectAnchors: Confidence-Based Anchor Selection

```
Require: Window positions $\mathbf{P}_w$, Window features $\mathbf{F}_w$, Reference $(\mathbf{F}_r, \mathbf{G})$
Require: Number of anchors $K$
Ensure: Anchor indices $\mathcal{A}$, Target coordinates $\mathbf{T}$
$n \leftarrow |\mathbf{P}_w|$                                        ▷ Window size
// Compute confidence for each frame in window
for $i = 1 \ldots n$ do
    $j_i \leftarrow \arg\min_j \|\mathbf{P}_w[i] - \mathbf{G}[j]\|$              ▷ Nearest reference tile
    $c_i \leftarrow \langle \mathbf{F}_w[i], \mathbf{F}_r[j_i] \rangle$                  ▷ Cosine similarity
end for
// Select top-K highest confidence as anchors
$K' \leftarrow \min(K, n)$                                  ▷ Handle small windows
$\mathcal{A} \leftarrow \textsc{ArgTopK}(\{c_i\}_{i=1}^n, K')$
// Get target coordinates for anchors
$\mathbf{T} \leftarrow \{\mathbf{G}[j_i] : i \in \mathcal{A}\}$
return $\mathcal{A}, \mathbf{T}$
```

Algorithm A4 OptimizeSE2: Local Rotation and Translation Optimization

```
Require: Window positions $\mathbf{P}_w$, Anchor indices $\mathcal{A}$, Anchor targets $\mathbf{T}$
Require: Window features $\mathbf{F}_w$, Reference $(\mathbf{F}_r, \mathbf{G})$
Require: Coarse steps $N_c = 12$, Search range $\theta_{max} = 0.2$ rad
Ensure: Refined positions $\mathbf{P}'_w$
function SCORE($\theta$)
    $R \leftarrow \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$
    $\mathbf{P}_{rot} \leftarrow R \cdot \mathbf{P}_w^\top$                          ▷ Rotate positions
    // Compute translation from anchor residuals
    $\mathbf{t} \leftarrow \text{median}_{i \in \mathcal{A}}(\mathbf{T}[i] - \mathbf{P}_{rot}[i])$
    $\mathbf{P}' \leftarrow \mathbf{P}_{rot}^\top + \mathbf{t}$
    // Evaluate similarity at transformed positions
    $s \leftarrow 0$
    for $i = 1 \ldots |\mathbf{P}_w|$ do
        $j \leftarrow \arg\min_j \|\mathbf{P}'[i] - \mathbf{G}[j]\|$
        $s \leftarrow s + \langle \mathbf{F}_w[i], \mathbf{F}_r[j] \rangle$
    end for
    return $s$
end function
// Coarse rotation search
$\Theta_c \leftarrow \{-\theta_{max} + \frac{2\theta_{max} k}{N_c} : k = 0, \ldots, N_c - 1\}$
$\theta_c^* \leftarrow \arg\max_{\theta \in \Theta_c} \text{SCORE}(\theta)$
// Fine rotation search
$\theta^* \leftarrow \text{BRENTMINIMIZE}(-\text{SCORE}, [\theta_c^* - 0.2, \theta_c^* + 0.2])$
// Apply optimal transformation
$R^* \leftarrow \begin{bmatrix} \cos\theta^* & -\sin\theta^* \\ \sin\theta^* & \cos\theta^* \end{bmatrix}$
$\mathbf{P}_{rot} \leftarrow R^* \cdot \mathbf{P}_w^\top$
$\mathbf{t}^* \leftarrow \text{median}_{i \in \mathcal{A}}(\mathbf{T}[i] - \mathbf{P}_{rot}[i])$
$\mathbf{P}'_w \leftarrow \mathbf{P}_{rot}^\top + \mathbf{t}^*$
return $\mathbf{P}'_w$
```

Algorithm A5 AverageOverlaps: Overlap Consensus Averaging

```
Require: Position estimates from all windows (stored during refinement)
Require: Window size $W$, Overlap $O$, Trajectory length $N$
Ensure: Final positions $\mathbf{P}$
$\mathbf{S} \leftarrow \mathbf{0}_{N \times 2}$                          ▷ Sum of position estimates
$\mathbf{C} \leftarrow \mathbf{0}_N$                                ▷ Count of estimates per frame
$W_s \leftarrow W - O$                              ▷ Step size
// Accumulate estimates from each window
for $k = 0, W_s, 2W_s, \ldots$ while $k < N$ do
    $\mathcal{W} \leftarrow [k, \min(k + W, N))$
    $\mathbf{P}_w \leftarrow$ refined positions for window $\mathcal{W}$
    for $i \in \mathcal{W}$ do
        $\mathbf{S}[i] \leftarrow \mathbf{S}[i] + \mathbf{P}_w[i - k]$
        $\mathbf{C}[i] \leftarrow \mathbf{C}[i] + 1$
    end for
end for
// Compute average
for $i = 1 \ldots N$ do
    $\mathbf{P}[i] \leftarrow \mathbf{S}[i] / \mathbf{C}[i]$
end for
return $\mathbf{P}$
```

References

1. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
2. Arandjelović, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
3. Keetha, N.; Mishra, A.; Karhade, J.; Jatavallabhula, K.M.; Scherer, S.; Krishna, M.; Garg, S. AnyLoc: Towards Universal Visual Place Recognition. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3082–3089.
4. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; Le, Q.V.; Adam, H. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
5. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332.
6. Workman, S.; Souvenir, R.; Jacobs, N. Wide-Area Image Geolocalization with Aerial Reference Imagery. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3961–3969.
7. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 2990–2999.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.