BEVLoc: Cross-View Localization and Matching via Birds-Eye-View Synthesis

Christopher Klammer*†
Carnegie Mellon University

Michael Kaess*
Carnegie Mellon University

Abstract— **Ground to aerial matching is a crucial and challenging task in outdoor robotics, particularly when GPS is absent or unreliable. Structures like buildings or large dense forests create interference, requiring GNSS replacements for global positioning estimates. The true difficulty lies in reconciling the perspective difference between the ground and air images for acceptable localization.**

Taking inspiration from the autonomous driving community, we propose a novel framework for synthesizing a birds-eye-view (BEV) scene representation to match and localize against an aerial map in off-road environments. We leverage contrastive learning with domain specific hard negative mining to train a network to learn similar representations between the synthesized BEV and the aerial map.

During inference, BEVLoc guides the identification of the most probable locations within the aerial map through a coarse-to-fine matching strategy. Our results demonstrate promising initial outcomes in extremely difficult forest environments with limited semantic diversity. We analyze our model's performance for coarse and fine matching, assessing both the raw matching capability of our model and its performance as a GNSS replacement.

Our work delves into off-road map localization while establishing a foundational baseline for future developments in localization. Our code is available at: https://github.com/rpl-cmu/bevloc

I. INTRODUCTION

In the realm of navigation, the presence of reliable and consistent GPS is paramount in existing localization solutions to help constrain the pose optimization to mitigate drift. However, GPS often is disrupted by interference from excessive forestry, overarching structures, atmospheric conditions, or adversaries intentionally jamming GPS signals. When GPS is absent, visual odometry (VO) and visual-inertial odometry (VIO) systems will eventually drift and become globally inaccurate without any sort of global reference or registration [4] [8]. These inaccurate estimates induce risks such as entering intraversable areas, diverging from the planned path, and damage to the environment or the robot itself.

The challenges and difficulties of state estimation are only exacerbated by using only vision sensors without the use of a reliable GPS signal. Though deploying high cost LiDAR sensors may delay the presence of drift, it is still inevitable without re-visitation or loop closures [21][27]. To combat

* C. Klammer and M. Kaess are part of the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213
Email: {cklammer, kaess}@cs.cmu.edu
† C. Klammer is part of KEF Robotics, Pittsburgh, PA, 15206
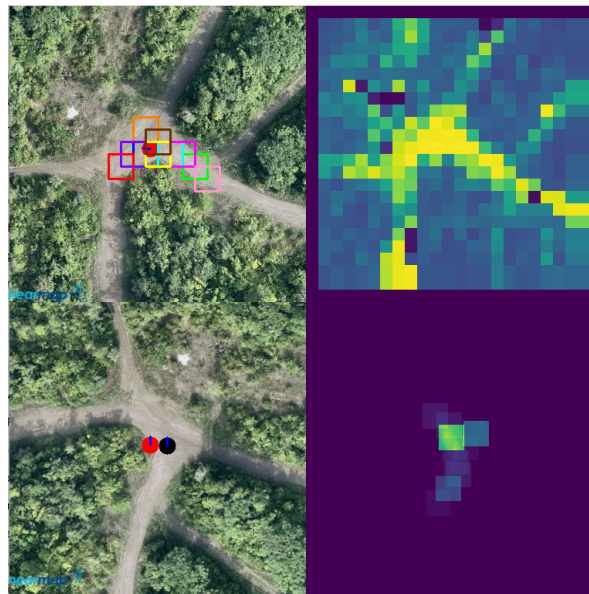Email: chris@kefrobotics.com

Fig. 1. Visualization of the coarse to fine matching strategy used by BEVLoc. The top panes show the coarse matches with the map aligned image and the corresponding correlation volume. The bottom panes show the map rotated by the predicted yaw angle and the refinement of the coarse location to create a probabilistic prediction for the localization by weighting the correlation maps over the top k coarse matches. The red circle denotes the ground truth localization, the black circle denotes the predicted localization.

this, we explore using a prior map of the environment to aid our GPS-denied localization pipeline by visually matching against and attempting to re-localize within the known map.

While many existing methods focus on rich semantic content present in on-road datasets, our focus rests in scenarios with sparse semantic information. The main challenge lies in harvesting whatever information possible to create geometric and semantic features to match against. However, this comes with challenges in the form of consistency between the aerial map and first-person-view (FPV) images - seasonal changes and change in vegetation can add additional complexity.

We seek to solve the "kidnapped robot problem" for an autonomous ground robot given only a map of the environment, a prior GPS location, and the vision sensors onboard the robot. Our motivation lies in focusing on localization in unstructured and off-road environments. Our work proposes a vision-centric pipeline to learn similar representations between aerial maps and ground camera images in order to localize the robot within the map. Using the semantics and geometry of the scene, our method creates a synthetic birds-
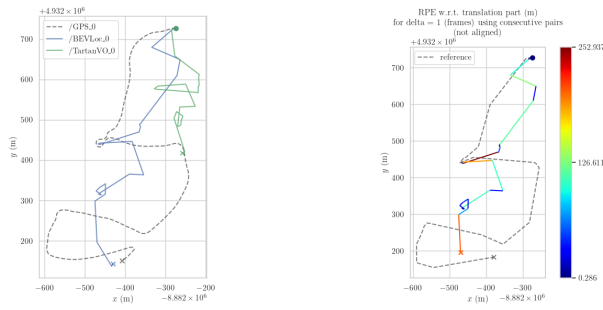
Fig. 2. Illustration of our method's performance, leveraging registration estimates to mitigate drift from visual odometry. **Left:** Comparison of BEVLoc and Tartan VO against the ground truth GPS trajectory. **Right:** RPE performance of BEVLoc throughout a long trajectory against GPS ground truth.

eye-view (BEV) representation to reconcile the perspective difference between aerial and the ground images.

Our contributions include a contrastive learning framework to provide an embedding for aerial and ground images to be used for downstream matching. We propose a coarse-to-fine matching approach to fuse a registration estimate with the sensor data in order to localize within an aerial map.

II. RELATED WORKS

Ground-to-air matching is not a new task, oftentimes, existing works have treated this as an image retrieval problem for georegistration. Many existing works have looked at cross-view localization as an image retrieval problem for geolocalization. These methods showed initial promise, using generative models to bridge the gap between ground and aerial imagery to help learn more robust feature descriptors while utilizing contrastive learning to further close the domain gap [17]. Other works have focused primarily on cross-view correspondence, encoding semantic correspondences over time to be trained in a contrastive manner against reference satellite imagery embeddings [22].

Additionally, recent works have explored the potential of lifting sensor information from the ground for perception. Initial works looked at lifting multi-view camera images to a shared birds-eye-view representation (BEV), maintaining visual features from cameras and learn depth priors [7] [16]. Often, works combine these visual features with geometric features from LiDAR and/or radar in order to maximize perception accuracy for downstream birds-eye-view feature maps in the same detection and segmentation tasks. These methods use LiDAR to model the occupancy of the scene, creating geometric features and using splatting operators or pure projection of feature maps to create a shared representation [2] [10] [11] [13] [26]. Notably, SimpleBEV [5] took a closer look, analyzing what strategies made the biggest difference for perception tasks, highlighting the utility of bilinear interpolation along rays instead of predicting monocular depth priors. These methods focus on perception in a fixed three-dimensional grid around the robot ego-frame and infer the localization of objects rather than the robot itself.

We take inspiration from these works to lift image features to a birds-eye-view, using cameras available onboard and relying on stereo depths as a depth prior to better constrain the geometry of the scene.

Most similar to our work, [20] [19] have looked at 3DoF localization given multi-view images at an instantaneous point in time for widely available public street maps. These works focus on the strength of the semantics for localization and learning depth priors for neural maps. Their pose alignment strategy demonstrates the potential of BEV featurization and its accuracy in 3DoF visual positioning and semantic mapping given only a single image. As well, they look towards creating a contrastive learning framework as well, using pose supervision and mining high likelihood false positives through RANSAC adjacent strategies. On top of this and focusing on extremely hard negative samples, [18] provides well suited guidance on mining said negatives for contrastive learning frameworks to create representations for our usage.

Lastly, visual place recognition has received a resurgence with the introduction of foundational models [15]. Their matching capability has been demonstrated for visual place recognition and has been shown to work across a diverse set of scenes [1] [9]. More recent works have shown the utilization of visual place recognition on a real aerial robotics platform on real datasets [22]. This furthers our confidence in integrating foundational models for matching on real robotics platform and on state estimation pipelines.

III. METHODOLOGY

A. Problem Setup

Given a prior 3DoF unmanned ground vehicle (UGV) pose $P_0 = (x, y, \theta_{yaw})$, our method seeks to infer a global state estimate using vision sensors alone. We utilize the Tartan Drive 2.0 dataset [23][24] for all of the real-world UGV data. This includes a Carnegie Robotics MultiSense S21 sensor with FPV images and depth at 10Hz. From these sensors, we aim to create a synthesized BEV feature map BEV_G can be used to match against cropped regions of the aerial maps, encoded as feature maps BEV_A. We utilize TartanVO[25] to estimate local odometry $(\Delta x, \Delta y, \Delta \theta)$ between consecutive frames capture local motion of the robot, aiding in building a temporally informed representation for our method. Ultimately, we seek to match a ground representation BEV_G against multiple "aerial crops" or cut out locations centered at a GPS location. These representations, denoted as BEV_A, provide a global state estimate to re-localize the robot within the map and to correct for drift in the robot's trajectory. As for assumptions:

1) We assume the sensors are calibrated and the intrinsics and extrinsics are all known.
2) We assume the map is of known resolution and orthorectified.
3) We are given purely the map images and the sensors on-board the UGV. There are no known semantic images to simplify the learning process.
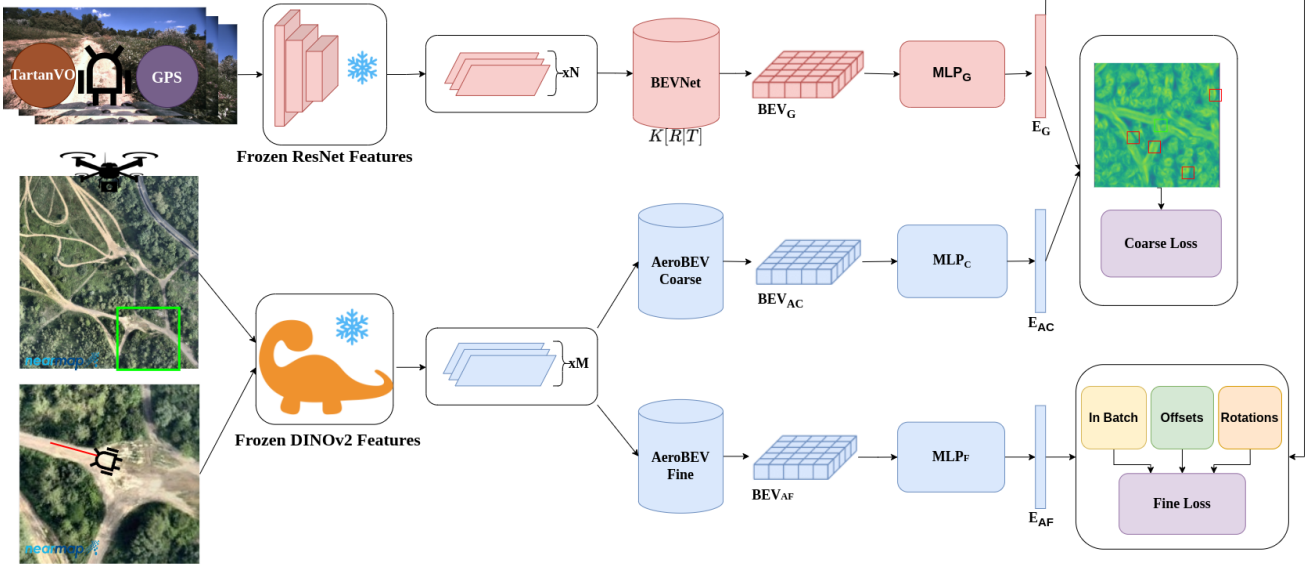
Fig. 3. BevLoc Contrastive Learning Training Pipeline. Feature maps are encoded from ground and aerial camera images. The ground features are lifted to 3D to create a semantic and temporally consistent BEV feature map to be compared against the aerial feature map. The embeddings are created and used to find hard negatives near the prior location and prior rotation to learn how to match ground to aerial images.

B. Lifting Features from the Ground

Our first major task is to lift camera images on the ground to compare against semantics in the aerial image. The top branch of 3 shows the process of processing camera images to be lifted into a BEV feature representation. For this encoding, we elect to use a ResNet-101 convolutional network [6] over foundational models. The reason for this lies in the stride size of 14 in the foundational backbones creates much lower resolution feature maps compared to ResNet-101, which could spawn ambiguities about the spatial locality of features. In turn, we theorize this could potentially decrease the localization accuracy of the features when projecting them to the feature volume.

We choose to place pixels of 2D feature maps into a 3D feature volume using the corresponding depth image to capture the visual features in a discretized grid around the robot. These features are placed at location X_{feat} in the corresponding voxel grid with voxel size of $[V_x, V_y, V_z] = [.3m, .3m, 3.0m]$ and grid size of $[G_x, G_y, G_z] = [32m, 32m, 16m]$. The transformation of features from camera to the grid, unprojecting the features followed by a translation to a grid frame such that the robot is in the middle of the grid as such that:

$$X^{feat} = T_{ned}^{grid} T_{cam}^{ned} \Pi(\begin{bmatrix} xz \\ yz \\ z \end{bmatrix}) \qquad (1)$$

This feature volume accumulates features, which are then averaged using the cumsum trick [16] to efficiently summarize features per voxel. Next, we elect to use max pooling over the pillars like [20][5] to create a geometrically consistent BEV

feature map $BEV_G^{(i)}$ for a given timestep i and highlight important semantics the map would reflect. We elect to use temporal information by utilizing the local odometry estimates to build a locate feature map and concatenate many BEV_G maps for a more rich semantic profile. We do this by using the last pose in the batch as the reference pose $T = [R|t] = [I|0]$ and each antecedent pose relative to the reference. A pose at time k is calculated as:

$$T_k^{odom} = \Pi_{i=ref}^{k} T_i^{-1} \qquad (2)$$

The final BEV feature map at a timestep k includes combining the features at every pixel, placing them into the grid then transforming the grid to respect the odometry.

$$BEV_k = T_k^{odom} X_k^{feat} \qquad (3)$$

The final map BEV_G concatenates the BEV feature maps for each time step channel-wise for a more semantically rich and temporally sensitive representation for the entire batch of size B, denoted by the concatenation operator $\bigoplus$.

$$BEV_G = \bigoplus_{}^{B} BEV_k \qquad (4)$$

Our method, much like other methods, [5][7][16] send this birds-eye feature map through a BEV encoder to compress and summarize the features for each spatial cell before passing through a linear layer to create the embedding E_G which is used for a representation to be compared against the aerial map.

C. Encoding the Aerial Images

1) Coarse Network: We train an AeroBEV coarse network using no prior rotation to receive a general localization

for the robot. We use a correlation mining strategy to help select hard negatives, discussed more in III-D.2

E_{AC} is produced by taking all of the map aligned cells and encoding them with a DINOv2 encoder. Since we look to capture coarse features rather than fine ones, DINOv2 is a strong backbone candidate for capturing the general semantics of the map cell and will likely be less prone to small discrepancies or noise in the map cell. This network is fairly simple, focusing on taking the feature map and feeding forward to an embedding using a linear layer, creating an embedding for each grid cell. The aim is to provide a descriptive embedding that can be compared to the sequence of recent ground images.

2) Fine Network: We train a separate AeroBEV network, denoted as AeroBEVFine, using the robot's prior rotation and aligning the aerial crop to be rotated around the robot's current position. Positives are denoted as high correlation matches with a rotation within $\tau_\theta = 10$ degrees of the robot and within $\tau_{far} = 3$ meters of the robot. We have separate mining modules to mine map locations are responsible for ensuring this network is neither rotation invariant nor translation invariant III-D.4. This network is identical to that of the coarse network besides the loss function which we discuss below.

E_{AF} is produced by taking all of the robot-aligned cells and encoding them with a DINOv2 or ResNet-101 encoder. That is, we use the current estimate θ_{yaw} to rotate the map and create robot-aligned embeddings. This network architecture is the exact same as the coarse network besides the encoder that is used. We select to use a smaller DINO-B encoder or ResNet-101 encoder to reduce the size of the features. Notably, we found that the ResNet-101 encoder received superior results to the foundational models, despite the strength in expressing scene semantics, we discuss this in more detail in IV-C.

D. Map Matching as a Contrastive Learning Problem

Given an embedding E_G for the ground images and an embedding E_A for the aerial crops centered around a GPS location, we now aim to minimize the distance between the embeddings for the last n camera images and the aerial crop at the given GPS location and yaw angle.

1) General Loss: We use a generalized cosine loss with a margin m which specifies the distance in the embedding space we look to have between positive samples and negative samples.

$$\mathcal{L}_+(...,y) = \frac{1}{|\mathcal{E}_A+|} \sum_{E_A \in \mathcal{E}_{A+}} y \cdot d(E_G, E_A) \qquad (5)$$

$$\mathcal{L}_-(...,y,m) = \frac{1}{|\mathcal{E}_A-|} \sum_{E_A \in \mathcal{E}_{A-}} (1-y) \cdot (m - d(E_G, E_A)) \qquad (6)$$

2) Mining Negative Coarse Aerial Embeddings: We define the coarse problem over the aerial crop of the map around the ground truth GPS location. We extract a local map of size $[A_x, A_y] = [384pix, 384pix]$ and utilize the set of all non-intersecting aerial embeddings $\mathcal{E}_A$ as our samples for the contrastive learning problem. During inference, this formulation changes slightly, with the most recently inferred pose of the robot being the center of the local map.

We define each region that creates an aerial embedding as the size of the ground robot's local map $[G_x, G_y]$. At training time, we choose only one positive to be the map cell in which the robot is closest to the center with the rest being negatives. With only one positive, we aim to use the general semantics in the cell to learn coarse localization predictions to be refined in the subsequent modules.

When mining negatives, we take each coarse embedding at the current time step and find the correlation with the ground embedding by taking the dot product. We mine negatives as any sample with: $corr(E_G, E_A) \geq 0$. As negatives become more difficult to mine, we establish an additional criteria that search for the top 25 percent of false positive correlation values and mark those as negatives. Therefore, we will always have, at least, 25 percent of the current map cells as negatives. We choose to not pass in all of the map cells to focus on learning useful areas to disassociate from the ground embedding while not hampering the network by overfitting on easy examples.

3) Coarse Loss: We define the coarse loss over the set of all non-intersecting aerial embeddings $\mathcal{E}_A$ and calculate their distance from the ground embeddings. We penalize the positive sample by its distance from the ground embedding. Meanwhile, we enforce a constant margin m_c as a target distance for negative samples, encouraging negatives to be pushed as far away as possible. Empirically, we set $m_c = 1.5$.

$$\mathcal{L}_c = \mathcal{L}_+(y) + \mathcal{L}_-(...,y,m_c) \qquad (7)$$

4) Mining Negative Fine Aerial Embeddings: Our goal in the fine problem is to refine the coarse matches in order to receive high quality global localization estimates. Our method method samples offsets in a random direction with offset value samples from a uniform distribution, making it equally likely we will sample a positive or a negative near the ground truth location. We take each of these samples and accept them as positives if they are within a threshold τ_d.

$$x_o, y_o \sim U[0, \sqrt{2} \cdot \tau_d] \qquad (8)$$

$$y_A = || \begin{bmatrix} x_o & y_o \end{bmatrix} ||_2 < \tau_d \qquad (9)$$

Likewise, we define rotations that are greater than 10 degrees to be negatives, taking care to handle angle wraparound. To mine the rotation negatives, we sample n_{rot} angles and take their chips at the ground truth location to produce embeddings that are not invariant to fine rotations.

5) Fine Loss: For the fine loss, we have three separate components:

1) An offset loss, which penalizes nearby negatives and penalizes hard positives by a Gaussian weight proportional to the physical distance from the ground truth location.

2) A rotation loss, which discourages overly rotated aerial crops at the given location.

3) A within-batch loss, which assigns poses within the batch positive or negative relative to the last pose

When performing fine matching, it is desirable for our embeddings to have a sense of distance when the robot is at different rotations. To reflect this, we introduce a loss that pushes away map embeddings that are at acceptable localization but have an inaccurate heading for the robot. $m_R = 0.8$

$$\mathcal{L}_R = \mathcal{L}(..., y_R, m_R) \tag{10}$$

For the offset aerial embeddings, we introduce additional Gaussian penalty loss $\mathcal{L}_G$ that artificially pushes away samples more the further they are from the ground truth location within the map

$$\mathcal{G}(d, \sigma) = e^{-\frac{1}{2} \cdot (\frac{d}{\sigma})^2} \tag{11}$$

We use RANSAC to sample a set number of positive and negative samples around the ground truth location. The final loss calculates a Gaussian weighted loss that takes into account the similarity to their embedding in addition to the physical distance d_p away from the ground truth location. We set $\sigma_o = 64$ pixels and $m_o = 1.5$

$$\mathcal{L}_O = \mathcal{G}(d_p, \sigma_o)\mathcal{L}(..., y_o, m_o) \tag{12}$$

The within-batch loss checks for samples that are within a certain distance threshold τ_{far}. In our experiments, τ_{far} is 3 meters and m_B is 1.25.

$$y_B^{(i)} = d(X_i, X_{ref}) <= \tau_{far} \tag{13}$$

$$\mathcal{L}_B = \mathcal{L}(..., y_B, m_B) \tag{14}$$

E. Coarse to Fine Matching

During the inference phase, we employ a heuristic coarse-to-fine matching strategy to harness the fine network. This strategy consists of:

1) Identification of the top k highly correlated matches and generation of a local correlation volume C_θ for a given yaw angle θ.
2) Outlier rejection by retaining high correlation estimates and employing multiple different orientations of the aerial crop to eliminate outliers.

To construct C_θ we scan match around the coarse match for $x \in [-\frac{G_x}{2}, \frac{G_x}{2}, s_x], y \in [\frac{-G_y}{2}, \frac{G_y}{2}, s_y]$ where $s_x = s_y = 1$ meter to find the highest correlation estimates. We weight these k correlation volumes, choosing to weight the top 3 coarse matches' contribution to be 70 percent of the correlation volume. Our assumption here is during our training process, we learned to lower the correlation values of nearby locations and will have a defined peak for the most probable estimate. However, this is very prone to outliers as some regions may look extremely similar to other regions in the map. Therefore, we propose looking at multiple orientations of map locations, targeting high correlation gradients across different angles. Notably, our work selects $\theta \in [-20, 20, s_\theta]$ with the positive being $\theta = 0$ and the other orientations being negatives.
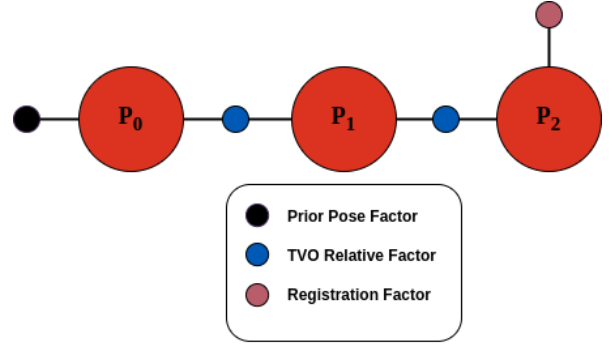

Fig. 4. An illustration of the global pose graph which corresponds to the 3DoF pose of the vehicle. As a vision-only pipeline, we utilize relative visual odometry measurements and high quality registration estimates.

We use the final fused correlation maps as our measurement covariance for the state estimate. Noise in the coarse and fine matching will be reflected in this covariance and is calculated by the outer product of the distance from the predicted location μ to the query location x_i, and weighted by a normalized probability for the cell p_i

$$\Sigma_r = \sum_i p_i(x_i - \mu)(x_i - \mu)^T \tag{15}$$

F. Pose Estimation via Non-Linear Optimization

We employ a framework for non-linear pose optimization using a factor graph [3]. This approach allows us to refine estimates from multiple streams of information.

Our factor graph looks to solve a non-linear least squares optimization where we estimate all poses up to a time t and perform a registration estimate every n keyframes as seen in Fig. 4. We optimize the series of poses denoted as:

$$\mathcal{X}_t = \{P_0, P_1, ..., P_t\} \tag{16}$$

$$\mathcal{X}_t^* = \text{argmin}_{\mathcal{X}_t} ||x_o||_{\Sigma_0}^2$$
$$+ \sum_i^t (||P(x_{i-1}, x_i)||_{\Sigma_T}^2)$$
$$+ \sum_j^{t/n} (||R(x_j)||_{\Sigma_r}^2) \tag{17}$$

The initial pose factor is initially the last received GPS measurement to better constrain the optimization. Additionally, we utilize the sensors available aboard the UGV, utilizing Tartan VO[25] relative pose factors to reflect the local motion of the robot. We utilize a semi-open loop system, only using GPS readings as a prior estimate for the current at a rate of 0.1Hz to analyze how our system deals with intermittent GPS signal, otherwise, the robot uses the previous optimization result as the initial guess for the pose estimate. Additionally, the registration estimate is used so long as the match is deemed to not be an outlier by the matching module. We add our method's registration estimate at a rate of 5Hz. We solve the optimization with the Levenberg–Marquardt algorithm [14].

IV. RESULTS

A. Dataset

For all of our experiments, we utilize the Tartan Drive 2.0 real world dataset which lauds over seven hours of on-road data with images, depth, RGB BEV images, IMU data, and much more [24][23]. The data rates for images are 10Hz, GPS at 50 Hz, and IMU at 100 Hz. In line with training our network, we preprocess color images and depth images to be the same size and scale the intrinsics. We also consider time synchronization of the different sensors to validate correct operation. For our experiments, we select 15 trajectories for training, 2 for validation, and 3 for testing. To emphasize the generalization capability of our method, we highlight the trajectories for the training and testing splits in Fig. 5
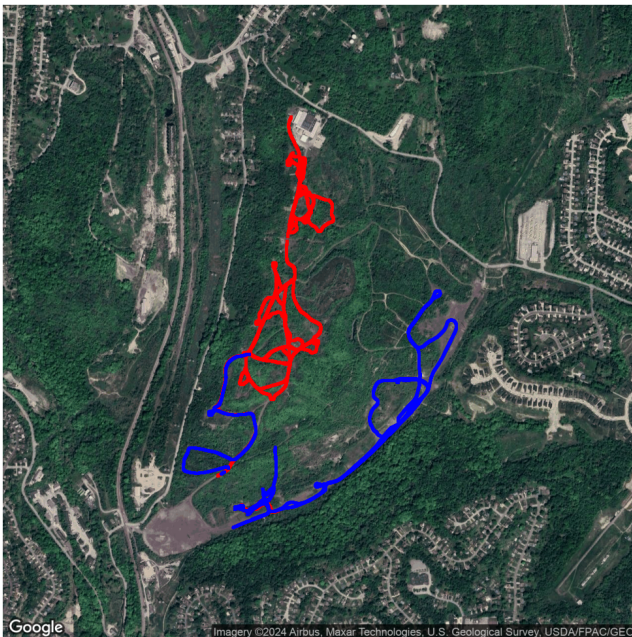

Fig. 5. Training and testing splits on TartanDrive 2.0 traced in Google Maps. Training trajectories are in red and testing trajectories are in blue with less than five percent trajectory overlap.

B. Metrics

1) Coarse Matching: For coarse matching, we analyze the recall, we extract a subset of the map around the last estimated location of size $[S_{crop}, S_{crop}]$ into disjoint and equally sized square map cells of size $[S_{cell}, S_{cell}]$. In our work, $S_{crop} = 378$ and $S_{cell} = 32$. For this calculation, we use the ground truth GPS to accurately evaluate how strong the coarse matching is.

We define the size of the BEV local map and size of the grid cell to be the same to simplify the matching process. We define the predicted location as the most probable k grid cells. A true positive is denoted as one of the k predictions landing in the same grid cell as the ground truth and a false positive for landing in any other cell. We analyze this metric from a granularity from $k = 1$ to $k = 10$. These coarse matches act as a downstream prior for our localization

refinement. This metric is crucial as it provides a strong prior for the robot's localization.

2) Fine Matching and Trajectory Error: We calculate the RMSE match from the ground truth position while running as a semi-open loop system. As noted earlier, this semi-open loop system models an intermittent GPS signal at a low frequency. We analyze the relative pose errors given this intermittent GPS signal. This allows us to take into account the quality of the global state estimate towards the optimization and also how good the fine matching is performing individually.

C. Experimental Results

For coarse localization, we see substantially better results in using the larger foundational models for coarse localization. Our best result is the utilization of the DINOv2-G backbone for the aerial encoder with the ResNet-101 backbone for the ground encoder. It is clear from our results that the more powerful semantic capability from the foundational models reflects improved performance, even though ResNet feature maps have a higher resolution. Fine matching performs well when coarse estimates provide a strong prior for matching. In line with expectations, ResNet performs strongest at top 1 recall but dips in performance once more matches are considered. This is likely because ResNet has higher-resolution feature maps, making geometric matching easier but provides less adept at semantic matching. We see a direct correlation between the localization error and the recall for the given backbone. A visualization of our matching results is present in Fig III-E

Overall, our method corrects trajectories without reliable GPS signals. The trajectory error error how significant improvement over VO alone. In most cases, the formulation results in the robot staying within the bounds of the trail, using the global state estimate to re-localize when drifting occurs. Although not quite as smooth as the reference trajectory with GPS state estimates, there is still a general success in the resulting localization as in 2. Interestingly, the ResNet-101 model minimizes the trajectory error and the fine-matching error. This speaks to the value of higher-resolution feature maps for fine-matching tasks, even if performance is diminished for coarse matching.

One failure case involves the registration of global state estimates with low covariance but matches against a hard positive. One example of this is similar road structures in unstructured environments. Also, difficult cases where the map and the ground images differ in semantics are challenging. One example encountered in the data was the presence of shadows in the aerial image, making matching extremely unlikely. Situations such as these suggest the need for a multi-modal approach to build robustness in the system. We see the introduction of IMU data, range data, and additional outlier filtering as a necessary step to remedy these issues. We leave further optimization, matching, and outlier rejection for future work.

Method	R@1	R@3	R@5	R@10
RGB Local Map Registration[12]	8.25	13.58	18.35	25.61
BEVLoc ResNet-101	**14.08**	29.21	41.07	59.78
BEVLoc DINOv2-B	13.91	31.41	44.18	61.07
BEVLoc DINOv2-L	12.77	29.29	**45.91**	61.32
BEVLoc DINOv2-G	13.48	**32.93**	45.66	**65.26**

TABLE II

ERROR (METERS) FOR FINE MATCH AND POSE ERROR ON
TARTANDRIVE 2.0 TEST SET - FIGURE 8 SEQUENCE

Method	RMSE Match	RPE
TVO [25]	-	90.45
BEVLoc ResNet-101	**59.80**	**28.23**
BEVLoc DINOv2-B	72.35	37.72
BEVLoc DINOv2-L	66.82	30.38
BEVLoc DINOv2-G	69.36	31.33

V. CONCLUSION AND FUTURE WORK

In our work, we presented a comprehensive framework for deriving global state estimates solely from vision sensors. By leveraging contrastive learning and employing rigorous methods for mining hard negatives, we have successfully learned representations between sequences of ground images and aerial maps. Our devised procedure for coarse-tofine matching utilizing the learned embeddings has shown promising results in unstructured environments, even with intermittent GPS signal availability.

While our approach demonstrates advancements, it is not without limitations. Challenges such as lighting variations, seasonal changes, and map updates persist, potentially impacting the robustness of our framework. While our framework provides a firm foundation, there is ample room for improvement.

Future advancements must aim to enhance geometric constraints to improve match quality and reduce reliance on GPS data. Furthermore, utilization of state-of-the-art attention-based matching or other learning techniques can further enhance the capability and robustness of the solution. Additionally, a refinement of alternate outlier rejection methods for registration state estimates and optimized marginalization strategies for non-linear optimization could increase the stability and robustness of the state estimations. By building upon our initial solution and addressing these challenges, we remain optimistic about the potential for significant advancements in the effectiveness and applicability of our framework for future research
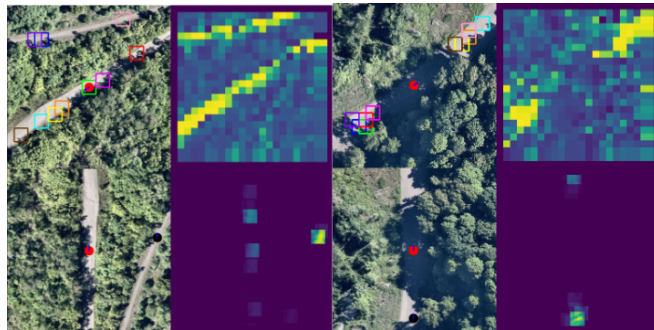
ACKNOWLEDGMENT

Fig. 6. Some failure cases encountered in coarse matching. The left exhibits similar road structure resulting in ambiguous matching. The right shows the difficulty of matching the ground image against shadows.

REFERENCES

[1] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition." In: *CVPR*. IEEE Computer Society, 2016, pp. 5297–5307. ISBN: 978-1-4673-8851-1.

[2] Yigit Baran Can, Alexander Liniger, Ozan Unal, Danda Pani Paudel, and Luc Van Gool. "Understanding Bird's-Eye View Semantic HD-Maps Using an Onboard Monocular Camera". In: *CoRR* abs/2012.03040 (2020). arXiv: 2012.03040.

[3] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. 2017. DOI: 10.1561/2300000043.

[4] Jeffrey Delmerico and Davide Scaramuzza. "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2502–2509. DOI: 10.1109/ICRA.2018.8460664.

[5] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. "Simple-BEV: What Really Matters for Multi-Sensor BEV Perception?" In: *arXiv:2206.07959*. 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. en. arXiv:1512.03385 [cs]. Dec. 2015.

[7] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. *FIERY: Future Instance Prediction in Bird's-Eye View from Surround Monocular Cameras*. en. arXiv:2104.10490 [cs]. Oct. 2021.

[8] Guoquan Huang. "Visual-Inertial Navigation: A Concise Review". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 9572–9582.

[9] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. "AnyLoc: Towards Universal Visual Place Recognition". In: *IEEE Robotics and Automation Letters* 9.2 (Dec. 2023), pp. 1286–1293.

[10] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. "PointPillars: Fast Encoders for Object Detection From Point Clouds". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12689–12697. DOI: 10.1109/CVPR.2019.01298.

[11] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers". In: *arXiv preprint arXiv:2203.17270* (2022).

[12] Yehonathan Litman, Daniel McGann, Eric Dexheimer, and Michael Kaess. "GPS-Denied Global Visual-Inertial Ground Vehicle State Estimation via Image Registration". en. In: *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 8178–8184. ISBN: 978-1-72819-681-7.

[13] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 2774–2781. DOI: 10.1109/ICRA48891.2023.10160968.

[14] J J More. "Levenberg–Marquardt algorithm: implementation and theory". In: (Jan. 1977).

[15] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

[16] Jonah Philion and Sanja Fidler. "Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D". In: *Proceedings of the European Conference on Computer Vision*. 2020.

[17] Krishna Regmi and Mubarak Shah. *Bridging the Domain Gap for Ground-to-Aerial Image Matching*. en. arXiv:1904.11045 [cs]. Aug. 2019.

[18] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. "Contrastive Learning with Hard Negative Samples". In: *International Conference on Learning Representations*. 2021.

[19] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. "OrienterNet: Visual Localization in 2D Public Maps with Neural Matching". In: *CVPR*. 2023.

[20] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. "SNAP: Self-Supervised Neural Maps for Visual Positioning and Semantic Understanding". In: *NeurIPS*. 2023.

[21] Tixiao Shan and Brendan Englot. "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4758–4765.

[22] Yujiao Shi, Xin Yu, Shan Wang, and Hongdong Li. "CVLNet: Cross-view Semantic Correspondence Learning fornbsp;Video-Based Camera Localization". In: *Computer Vision – ACCV 2022: 16th Asian Conference on Computer Vision, Macao, China, December 4–8, 2022, Proceedings, Part I*. Macao, China: Springer-Verlag, 2023, pp. 123–141. ISBN: 978-3-031-26318-7. DOI: 10.1007/978-3-031-26319-4_8.

[23] Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. *TartanDrive 2.0: More Modalities and Better Infrastructure to Further Self-Supervised Learning Research in Off-Road Driving Tasks*. en. arXiv:2402.01913 [cs]. Feb. 2024.

[24] Samuel Triest, Matthew Sivaprakasam, Sean J. Wang, Wenshan Wang, Aaron M. Johnson, and Sebastian Scherer. "TartanDrive: A Large-Scale Dataset for Learning Off-Road Dynamics Models". In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 2546–2552. DOI: 10.1109/ICRA46639.2022.9811648.

[25] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. "TartanVO: A Generalizable Learning-based VO". In: (2020).

[26] Chenyu Yang, Yuntao Chen, Haofei Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Y. Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. "BEVFormer v2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision". In: *ArXiv* (2022).

[27] Ji Zhang and Sanjiv Singh. "LOAM: Lidar Odometry and Mapping in Real-time". In: *Proceedings of Robotics: Science and Systems (RSS '14)*. July 2014.