GNSS-denied Ground Vehicle Localization for Off-road Environments with Bird's-eye-view Synthesis

Lihong Jin

CMU-RI-TR-24-54

August, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Michael Kaess, *chair*
Dr. Wenshan Wang
Easton Potokar

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family, friends,
and all who I had the honor to share this wonderful journey with.

Abstract

Global localization is essential for the smooth navigation of autonomous vehicles. To obtain accurate vehicle states, on-board localization systems typically rely on Global Navigation Satellite System (GNSS) modules for consistent and reliable global positioning. However, in real-world scenarios, GNSS signals can be obstructed by natural or artificial barriers, leading to temporary system failures and degraded state estimation for autonomous vehicles.

On the other hand, off-road driving presents unique challenges for unmanned ground vehicles (UGVs) due to irregular terrain, leading to unstable surfaces for traversal that affect the accuracy of state estimation. Dense forests or canyons can block GNSS signals, hindering precise absolute positioning. Additionally, visual odometry performance may suffer due to the lack of distinct and reliable features necessary for accurate state estimation.

To address these challenges, we propose a novel learning-based method that synthesizes a local bird's-eye-view (BEV) image of vehicle's surrounding area by aggregating visual features from camera images. The proposed model combines a deformable attention-structured network with an image rendering head to generate a BEV image. The synthesized image is then matched with an aerial map for cross-view vehicle registration in GNSS-denied off-road environments.

Our method overcomes the limitations of visual inertial odometry (VIO) systems and the substantial storage requirements of image-retrieval-based localization strategies, which are susceptible to drift and scalability issues.

Extensive real-world experimentation validates our method's advancement over existing GNSS-denied visual localization methods, demonstrating notable enhancements in both localization accuracy and registration frequency. Furthermore, our method effectively reduces VIO drifts when integrated with an on-board VIO system via factor graph optimization.

Acknowledgments

First, I would like to express my deepest appreciation to my advisor, Professor Michael Kaess, for his unwavering support and guidance throughout my master's studies. Our discussions on research, studies, and future directions in robot localization have been thoroughly enriching and inspiring. I appreciate his patience, willingness to listen, and dedication to offering support during challenging times. Thank you for providing the opportunity for this project that allowed me to learn and grow as a researcher.

I am grateful to my committee members, Doctor Wenshan Wang and Easton Potokar. Wenshan provided invaluable advice on this project and off-road autonomy, imparting extensive knowledge on robot learning in localization and mapping. I am grateful for her dedication to research and willingness to help. Easton and I collaborated closely on the same project, and I greatly appreciate his expertise in real-world systems, his patience, and team spirit.

Special thanks to Professor Kaicheng Yu for insightful discussions and valuable suggestions on BEV-related research. I also extend my gratitude to Prassanna and Jacob from NREC for their support in this work. Working with you was a delightful experience.

I am thankful to all of my lab members for their unreserved support and companionship over the years. Special thanks to Doctor Wei Dong, who has been a mentor and consultant since my second year in the MSR program. I learned a lot from him about research skills, planning, organizing, coding strategies, and stress management. My gratitude also goes to Ray for his input, knowledge and inspiring passion for research. I am grateful to Dan for his insightful research suggestions, paper recommendations related to my topics, and occasional advocacy during group meetings. I also thank Ananya for sharing her lab experience with me over the coffee chat; Easton, Taylor, and Ray for teaming up with me for course projects; and Tianxiang, Joe, Ray, Monti, and Dan for providing valuable feedback for my paper submission. Furthermore, I thank Ruoyang, Ray, Tianxiang, Andrew, Wei, Chunyu, Joe, Cindy, John, Akshay, and Taylor for sharing bike trips, tennis games, New Year's Eve, and pleasant weekend time with me. These experiences have been invaluable treasures and you all have made them so special.

My gratitude also goes to BJ and Dimi for their support in the continuation and successful completion of my MSR program. BJ offered valuable suggestions on how to cope with academic stress and explore all possibilities. I wish you the best for the future. Additionally, I appreciate Professor John Dolan for providing the TA opportunity and suggestions for advisor matching during my first semester at CMU.

I am very grateful to my beloved feline family members, Rum, previously Peanut, and Luna. During the long nights when I felt stressed and down, your presence offered immense comfort and companionship. I promise to always provide you with the best cat litter in the world :)

I am deeply grateful to my therapist, who has been supporting me over the past two years. Thank you for recognizing my struggles, helping me manage my chronic grief, loss, fear, anxiety, and healing for them. Your guidance helped me develop self-care, motivation, energy, confidence, and assertiveness for my work and life. Your dedication and compassion saved me more than once during my darkest times.

Lastly, I would like to thank my parents for supporting me in every way possible. Your constant encouragement is my guiding star, and your love and companionship are the vessels that allow me to explore the world.

Funding

This work was supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W911NF-20-D-0002. The content of the information does not reflect the position or the policy of the government and no official endorsement should be inferred.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 GNSS-denied Vehicle Localization

Global localization is a crucial component of the smooth navigation of autonomous vehicles. To obtain accurate vehicle states, on-board localization systems are typically equipped with GNSS modules to provide consistent and reliable global positioning. However, in real-world scenarios, GNSS signals can be obstructed by natural or artificial barriers, leading to temporary system failures and degraded state estimation for autonomous vehicles. Examples of GNSS-denied applications are shown in Fig. 1.2.

To achieve reliable localization in GNSS-denied environments, researchers have explored methods to maintain the robustness and precision of relative positioning, which is typically achieved by the integration of multiple sensors, e.g. inertial measurement unit (IMU) or inertial navigation system (INS), Light Detection and Ranging (LiDAR) and imaging cameras. Another approach is to perform absolute positioning, where vehicles are assumed to have prior knowledge about the environment, such as georeferenced data for local-to-global scan matching. One way of achieving this is to perform vision-based localization (VBL) with the help of visual place recognition (VPR), where autonomous agents investigate the environment with vision sensors such as an RGB camera, and continuously gather visual information to build a local representation of the surroundings, followed by matching it with prestored global representations, implicitly or explicitly. Various methods for VPR in urban scenarios have shown considerable success [2, 37, 44].

1.2 Off-road Autonomous Driving

In the realm of self-driving, off-road driving poses special challenges to UGVs due to several reasons: irregular terrain surfaces including mud, sand rocks, and water leads to unstableness of traversing vehicles, perturbing state estimation accuracy; natural obstacles such as trees, bushes can be difficult for vehicles to detect and navigate around, as shown in Fig. 1.1; dense forest or canyons may block GNSS signals for precise absolute positioning; furthermore, visual odometry performance may be degraded due to lack of distinct and reliable features for matching in state estimation.

1.3 Our Mission

In response to these challenges, we present a novel learning-based method that synthesizes a local BEV image of the surrounding area by aggregating implicit visual features from camera images. This approach integrates a modified BEVFormer [22] framework with a novel rendering head, employing template matching for precise cross-view registration between ground vehicles and aerial maps in GNSS-denied off-road environments.

We concentrate on the 2D relocalization of UGVs for non-urban settings bounded within defined areas. Equipped with trinocular RGB cameras and an Inertial Measurement Unit (IMU), the vehicle employs stereo VIO for state estimation. Our goal is to achieve accurate 2D positioning relative to a georeferenced aerial map, facilitating pose correction in the absence of GNSS signals, temporarily or persistently.

Previous study [24] has explored the creation of orthographic view images by accumulating geometric features over consecutive frames, coupled with Normalized Cross-correlation (NCC) for relocalization in a Global Positioning System (GPS)-denied situation. However, this approach is limited by the inherent drift of VIO systems, which can distort the accumulated geometric data, leading to inaccuracies in ground-to-air matching. Instead, we introduce a learning-based strategy for generating BEV images, using a Vision Transformer (ViT) [10]-based network for feature encoding. This method shows improved performance in generating local BEV images and supporting vehicle localization with georeferenced aerial maps.

Other research efforts [37, 42, 44] treat vision-based localization as an image retrieval problem, requiring substantial storage for on-board localization systems. In contrast, our approach generates local BEV images for direct template matching. This significantly reduces the need for extensive data storage, relying instead on a georeferenced map for real-time 2D localization.

1.4 Contribution

The proposed contains three main components: a feature encoder that maps visual representation from camera to top-down view, a rendering head that decodes learned features and renders top-down BEV images, and an image registration component for localization. An overview of our system is shown in Fig. 9.1. Our contributions can be summarized as follows.

- We propose a novel *learning*-based framework for ground vehicle localization that combines BEV image generation with *classical* template matching, eliminating the extensive dataset storage requirements of image-retrieval-based localization.
- We integrate the deformable attention module in [43] with the BEVFormer network, improving the encoding of features using offset networks [22], followed by an efficient image rendering head as a feature decoder capable of producing detailed top-down views of the local terrain.
- Through comprehensive experiments with real-world datasets, we demonstrate that our method exhibits more reliable localization accuracy and frequency compared to existing GNSS-denied visual localization techniques and generalizes to unseen trajectories.
- We integrate our map registration method with on-board VIO system and show our system reduces VIO drifts by a large margin.

Figure 1.1: An off-road driving scene

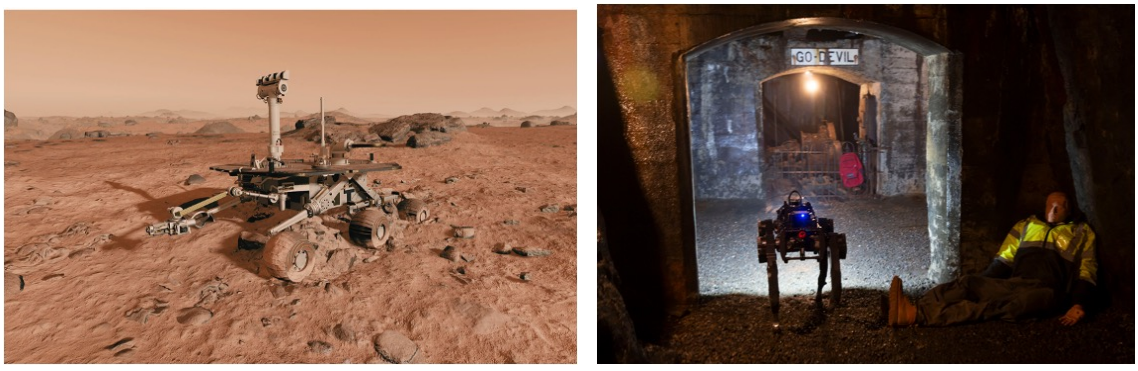

Figure 1.2: Applications of GNSS-denied localization (online sources)

Chapter 2

Background

2.1 Factor Graph for State Estimation

Estimation of an unknown robot pose $\mathbf{X}$ can be represented as a maximum a posteriori (MAP) problem given a set of sensor observations $\mathbf{Z}$. Applying Bayes' theorem, the posterior probability density can be represented with the product of the likelihood and the prior probability over the marginal likelihood. Given specific sensor observations, $p(\mathbf{Z})$ can omitted due to its irrelevance to MAP:

$$\mathbf{X}^{MAP} = \arg\max_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}) \tag{2.1}$$

$$= \arg\max_{\mathbf{X}} \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})} \tag{2.2}$$

$$= \arg\max_{\mathbf{X}} p(\mathbf{Z}|\mathbf{X})p(\mathbf{X}) \tag{2.3}$$

Using factor graph to represent the conditional probabilities and the relationship among sensor measurements and poses, the posterior probability can be represented with the product of factor potentials:

$$\mathbf{X}^{MAP} = \arg\max_{\mathbf{X}} \phi(\mathbf{X}) \tag{2.4}$$

$$= \arg\max_{\mathbf{X}} \prod_i \phi(\mathbf{X}_i) \tag{2.5}$$

Suppose all the sensor noise is Gaussian, factors will follow the form of:

$$\phi(\mathbf{X}_i) \propto \exp\left\{-\frac{1}{2}\left\|h_i(\mathbf{X}_i) - z_i\right\|^2_{\Sigma_i}\right\} \tag{2.6}$$

Taking the negative log and dropping term $\frac{1}{2}$, the problem can be converted to solving for a non-linear least-square problem:

$$\mathbf{X}^{MAP} = \arg\min_{\mathbf{X}} \sum_i \left\|h_i(\mathbf{X}_i) - z_i\right\|^2_{\Sigma_i} \tag{2.7}$$

By applying first-order Taylor Expansion to linearize target function, the problem can be converted to minimizing the Mahalanobis distance on the state update vector:

$$h_i(\mathbf{X}_i) = h_i(\mathbf{X}_i^0 + \Delta_i) \approx h_i(\mathbf{X}_i^0) + \mathbf{H}_i\boldsymbol{\Delta}_i \tag{2.8}$$

$$\boldsymbol{\Delta}_i = \mathbf{X}_i - \mathbf{X}_i^0 \tag{2.9}$$

The state vector can be updated until convergence with non-linear optimization methods, such as steepest descent, Gauss-Newton, Levenberg-Marquat, or Powell's Dogleg method:

$$\mathbf{X}_{i+1} = \mathbf{X} + \boldsymbol{\Delta} \tag{2.10}$$

2.2 Visual Inertial Odometry with Absolute Pose Correction

In robotic state estimation, IMU sensors provide acceleration and angular velocity measurements, which help to improve the precision and robustness of visual odometry (VO) when integrated with visual information, particularly in scenarios with high-velocity motions or low-texture environments where visual features may be difficult to track. However, VIO is prone to drift over time due to IMU sensors suffering from accumulative noise, bias instability, temperature sensitivity, etc; as well as cameras suffering from poor matching and inaccurate calibrations.

To mitigate drifts, VIO is usually used in combination with absolute pose cor-

rections, which are incorporated into the factor graph as additional constraints in a tightly coupled manner. These corrections can be derived from GPS measurements, known landmarks, or scan-to-map matching with pre-built georeferenced maps.

We represent the 2D registration results as prior factors and correlate them with the corresponding camera poses, shown as red dots in Fig. 2.1.

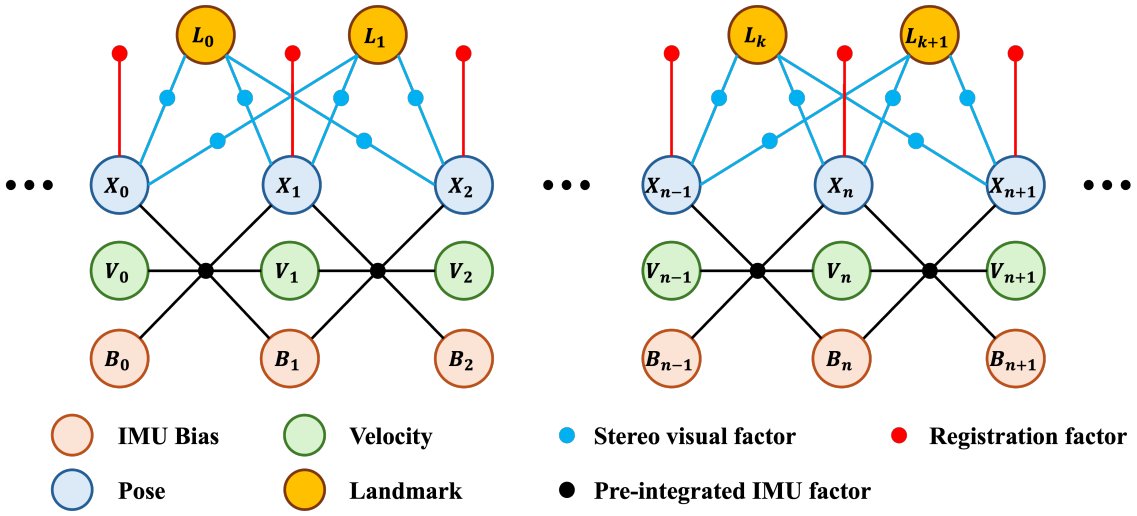

Figure 2.1: Factor graph for VIO with global constraints

2.3 Absolute Position Prediction with BEV Feature Learning

In predicting global pose from cross-view images, we consider a scenario where a vehicle, equipped with trinocular cameras and an IMU, is traversing flat natural terrain. A pre-stored aerial map of the area aids in localization. The vehicle's pose is predicted by the VIO system in a local coordinate frame as follows:

$$\mathbf{X}_t = \Big[x_t, y_t, \theta_t\Big] \in \mathbf{SE}(2). \tag{2.11}$$

We assume that the prediction for the azimuth angle θ_t from VIO is accurate, but the position estimates (x_t and y_t) may drift over time. Our system aggregates consecutive frames to construct a top-down representation of the environment for map registration.

Our system seeks to find the optimal pose prediction that minimizes the difference between camera feature representation and local aerial image:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \psi\Big(I'_{\text{bev}}(\mathbf{X}), I_{\text{map}}(\mathbf{X})\Big), \tag{2.12}$$

where ψ is a function to find $\mathbf{X}^*$ to achieve minimum distance between two representations, and provided by template matching in our system. I_{map} is the subset of aerial map with respect to vehicle pose, and I'_{bev} is the rendered BEV image given learned feature:

$$I'_{\text{bev}}(\mathbf{X}) = \varphi_{\text{render}}\Big(F_{\text{feat}}(\mathbf{X})\Big), \tag{2.13}$$

where φ_{render} is a mapping from the encoded feature F_{feat} to the resulting top-down BEV image, and is given by the rendering head in our model.

Chapter 3

Related Work

3.1 GNSS-denied Vehicle Localization

Vehicle localization in GNSS-denied environments can be broadly categorized into relative and absolute localization strategies. Relative localization aims to mitigate odometry drifts by fusing data from multiple onboard sensors with motion models, or by leveraging loop closures to correct drift relative to global frames [15]. Absolute localization, in contrast, involves constructing local maps from the vehicle's perspective and aligning them with a global georeferenced map to determine precise vehicle positions. Reference data for this process can vary, including High-definition (HD) maps [32], aerial satellite imagery, Digital Elevation Models (DEM) [19, 41], and OpenStreetMap (OSM) data [36]. While HD maps offer high accuracy, they are costly and data intensive. DEMs, primarily used for UAVs [41], cater to non-planar terrains and scale ambiguity, whereas OSM provides dense semantic and geometric details suitable for urban navigation. Aerial satellite maps present strong visual cues with detailed information for off-road localization.

Significant advancements have been made in aligning ground-level images with aerial imagery for localization. Viswanathan et al. [40] demonstrate effective ground-to-air image matching using satellite images by warping UGV panoramic images to a bird's eye view, comparing feature descriptors, and employing a particle filter for accurate localization. Based on this, recent work [24] focuses on generating an orthographic occupancy map by accumulation of local features and estimation of pose

through NCC, and optimizing the prediction of global pose through a registration graph [9]. In contrast, our approach adopts a Vision Transformer (ViT)-based [10] learning network to generate BEV images for ground-to-air matching, emphasizing frame-by-frame registration accuracy and reducing reliance on global trajectory optimization.

3.2 Learning Vision-based Localization

The evolution of vision-based localization has seen it conceptualized as an image retrieval task [42], employing contrastive learning to enhance the matching of onboard camera and satellite images [37, 44]. Efforts to improve image alignment include warping satellite imagery by polar transformation to match ground perspectives [44], and constructing semantic neural maps from camera images [37]. Further innovations leverage CNNs for feature extraction and BEV representation, enabling precise localization through 3D structure inference and matching [4, 11, 36, 45].

The advent of foundation models offers promising directions for Visual Place Recognition (VPR), demonstrating the adaptability of pre-trained models (e.g. DINO [6], DINOv2 [28]) to diverse environments without fine-tuning [18]. Subsequent work [13] integrates dense visual feature extraction with advanced filtering and global-local pose estimation via Extended Kalman Filters (EKF) for refined localization accuracy. Our methodology aligns with these advancements, utilizing a streamlined ViT architecture for efficient and accurate BEV image rendering and localization, minimizing parameter overhead while maximizing performance.

3.3 BEV for Autonomous Navigation

In the realm of self-driving applications, BEV related research has attract people's attention as a result of it's capability in representing top-down local environment from cross-view visual signals, such as data from Radio Detection and Ranging (Radar), LiDAR, monocular or multi-camera. Researchers have investigated BEV for 3D object detection [12, 17, 23, 38], semantic segmentation [29, 31, 35, 46], planning [26, 27], and mapping [5, 7, 8].

Recently, BEV representations [5, 20] have been enriched by encoding temporal and spatial features, as demonstrated by BEVFormer [22], which leverages attention mechanisms [21, 39, 43] for 3D object detection. Our work extends this concept by incorporating BEVFormer's feature propagation approach, ensuring our BEV representations integrate temporal information from successive frames. This strategy is complemented by recent explorations in temporal information encoding for BEV representation, highlighting the continuous evolution and application of these techniques in autonomous navigation [1, 3, 16, 33, 34].

Chapter 4

Learning Position Prediction

4.1 BEV Space and Reference Points

We define a 3D BEV space centered on the vehicle with a length of L, a width of W, and a height of H. The space is divided into $l \times w \times h$ grid cells, so that each cell represents a cubic size of $\frac{L}{l} \times \frac{W}{w} \times \frac{H}{h}$ in the real world. For each BEV grid, we sample one point in the center of the cubic and use the sampled points as 3D reference points, as shown in Fig. 4.1. 3D reference points are subsequently projected onto the camera frame given extrinsic information as a 2D reference point set. A visualization of 2D reference points is shown in Fig. 4.2. The 2D reference points are later used in extracting tokens for attention module.

The BEV query for attention module is a 3D trainable embedding with a dimension of $l \times w \times h$ representing the BEV space and serving as the query for deformable attention modules in the encoder. All intermediate BEV features in the network also follow the same spatial dimension. The specific range and dimension chosen for our experiment are described in Sec. 6.1.

4.2 Learning BEV Generation

Our proposed model aims at generating a top-down BEV map covering the 3D BEV space, centered around the vehicle at each time stamp. A diagram of our system is

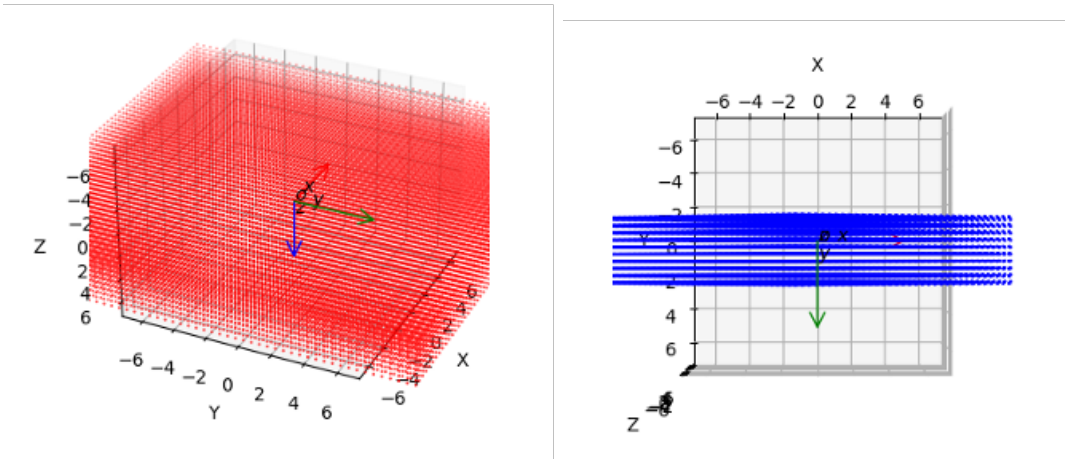

Figure 4.1: 3D reference points distribution
The left figure shows the side view and the right figure shows the front view

shown in Fig. 9.1.

During the training phase, camera images are patch projected and sent to the feature encoder (in blue) and rendering head (in orange) to generate BEV images (highlighted in yellow boxes). The aerial map image is rotated and cropped according to the GPS information provided, ensuring that the final label image accurately represents the BEV space surrounding the vehicle.

We utilize Mean Square Error (MSE) as the loss function to represent the difference between the ground truth aerial map and the learned BEV representation.

$$L_{\mathbf{MSE}}(\hat{y}, I'_{\text{bev}}(x)) = \frac{1}{N}\sum_{x=0}^{N}\left(\hat{y} - I'_{\text{bev}}(x)\right) \tag{4.1}$$

4.3 2D Localization with Template Matching

During the testing phase, the rendered BEV image is rotated according to the azimuth angle provided by the GPS, and matched against a local search region surrounding the vehicle position by NCC:

$$R(x,y) = \frac{\sum_{x',y'}(T(x,y)\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x,y)^2\cdot\sum_{x',y'}I(x+x',y+y')^2}} \tag{4.2}$$

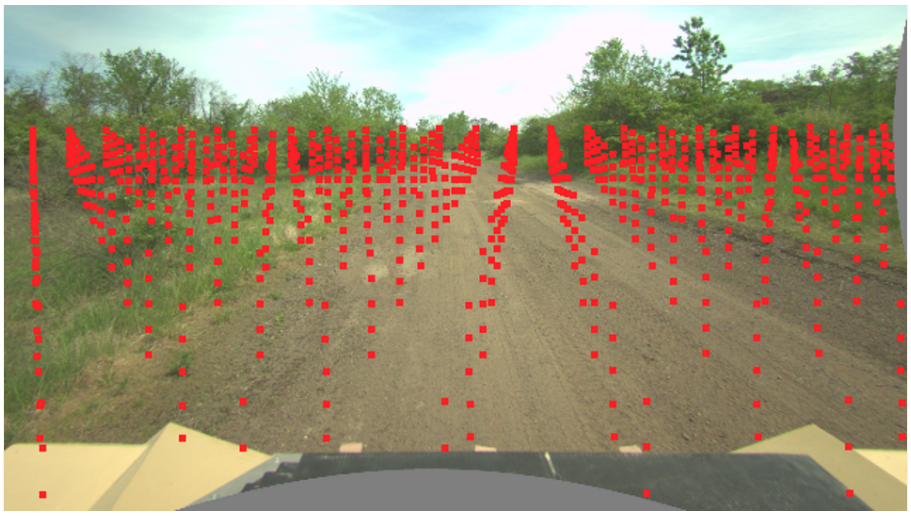

Figure 4.2: 2D reference points distribution on a camera frame

Chapter 5

BEVRender Architecture

5.1 Feature Encoding with BEVFormer

Adopting BEVFormer's framework [22], we propagate consecutive frame features to capture temporal information. Within a temporal window of T seconds, n frames ($3 \times n$ images in a trinocular setup) are sampled. A detailed setting can be found in Sec. 6.1. Starting with the earliest frames, camera images I_t^{cam} are processed through patch projection, which is a convolutional layer in our implementation, to obtain camera feature F_t^{cam} and sent to the encoder together with the BEV query Q and previous BEV feature B_{t-1} to obtain the encoded BEV feature for current timestamp B_t. The encoding process consists of two stages: a temporal attention stage that takes in query Q and previous timestamp BEV feature B_{t-1} for deformable attention:

$$B_t^{\text{temp}} = \text{DeformableAttn}\Big(B_{t-1}, Q\Big), \tag{5.1}$$

followed by a spatial attention stage that takes in temporal output and camera feature F_t for deformable attention:

$$B_t^{\text{spatial}} = \text{DeformableAttn}\Big(F_t^{\text{cam}}, B_t^{\text{temp}}\Big), \tag{5.2}$$

B_t is then projected to the location of the subsequent frame as B_t' according to the movement of the vehicle given by the GPS information, using affine transformations

in $\mathbf{SE}(2)$ and bilinear interpolation:

$$\Delta\mathcal{X} = \Big[\Delta x, \Delta y, \Delta\theta\Big] = \mathcal{X}_t - \mathcal{X}_{t-1}, \tag{5.3}$$

$$\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta & \Delta x \\ \sin\Delta\theta & \cos\Delta\theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{bmatrix}, \tag{5.4}$$

$$B_t^{'}(x_t, y_t) = \text{BilinearInterp}(B_t(x_t, y_t)). \tag{5.5}$$

Subsequently, $B_t^{'}$ serves as a query to the encoder together with the next camera feature F_{t+1} to obtain B_{t+1}. The propagation continues in the temporal window until we obtain the latest timestamp feature B^T. A diagram of temporal propagation is shown in Fig. 5.1. For each timestamp, we sample n frames from past T seconds, composing a training sample of $n+1$ camera frames together with current timestamp frame. Staring with the earliest timestamp in the window, BEV query Q is used to query camera feature F to obtain BEV feature B, which is subsequently projected to next timestamp vehicle position given GPS outputs, to obtain new feature B'. Propagation continues until the latest frame is processed. It should be noted that B_{t-1} is the same as query Q for the first frame in the temporal window:

$$B_{t-1} = Q \quad \text{if} \quad t = 0. \tag{5.6}$$

Unlike BEVFormer, our encoder simplifies to a single layer, totaling 1.44 million parameters, while supporting effective feature learning for downstream localization tasks. The architecture of the encoding layer is shown in Fig. 9.2, and the ablation study of the number of layers can be found in Table 6.3.

5.2 Deformable Attention Vision Transformer

The model architecture for our BEV feature encoder is shown in Fig. 9.2.

In contrast to BEVFormer that employs Deformable DETR [47], our approach

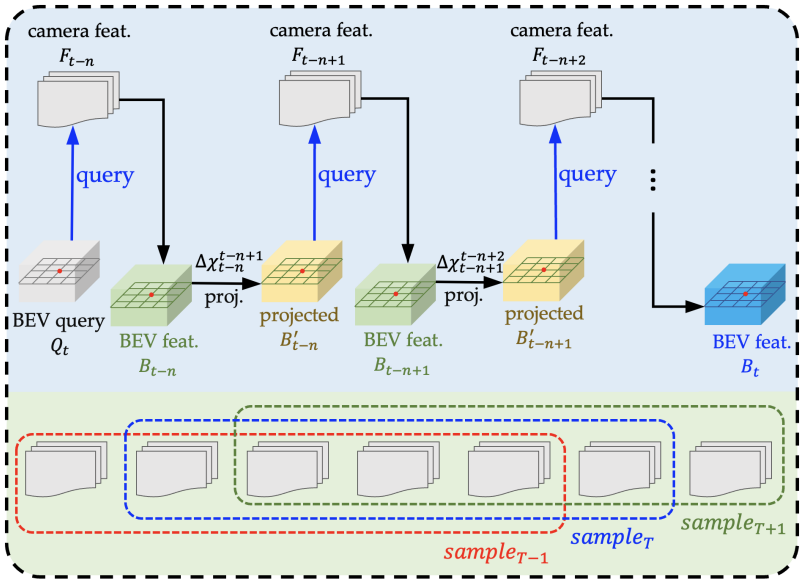

Figure 5.1: Temporal feature propagation and dataset organization.

utilizes the deformable attention [43], which uses offset networks to calculate adjustments to each reference point. The offsets are processed by an additional convolution layer θ_{offset}, as shown in Fig. 9.2, and its output modifies the original reference point to generate deformed reference points.

For spatial attention, offsets θ^i_{offset} are added to the reference points unique to each camera view i, acting as adjustments to the pixel locations of reference points. Consequently, we employ three distinct convolution layers dedicated to learning offsets as an adaptation to the trinocular system setting. The final output of the spatial attention layer is a stacking of features from three camera views, undergoing another convolutional layer to maintain the same spatial dimension as the BEV query and BEV features.

The output of deformable attention heads is formulated as

$$z^{(m)} = \sigma\Big(\frac{q^{(m)}{k^{(m)}}^\top}{\sqrt{d}} + \phi(B;R)\Big)v^{(m)}, \tag{5.7}$$

where q, k, v constructs the standard transformer attention [39] with softmax activation σ and scale normalization $\sqrt{d}$, enhanced by relative positional bias [25] in $\phi(B;R)$. A more detailed description of deformable attention formulation can be found in [43].

5.3 BEV Image Rendering Head

The BEV image rendering head is designed to translate encoded features into interpretable top-down views of the vehicle's surroundings. It is a straightforward convolutional neural network (CNN) architecture that takes as input the encoded BEV features with dimensions of $d \times l \times w$, where d is the model embedding dimension. Through a series of convolutional and upsampling layers, the BEV features are processed to generate a colored image of certain size, which serves as a top-down visual representation of the BEV space around the vehicle. The rendering head ensures that the resulting BEV image retains critical spatial information required for ground-to-aerial vehicle localization in GNSS-denied environments. The detailed structure of the rendering head is illustrated in Table 5.1.

Table 5.1: BEV rendering head architecture

block	layer
Decoder block 0	Conv2d + BN + ReLU
Decoder block 1	(Conv2d + BN)×4 + ReLU
Decoder block 2	(Conv2d + BN)×4 + ReLU
Decoder block 3	(Conv2d + BN)×4 + ReLU
Upsample block 0	Upsample + (Conv2d + BN)×2 + ReLU
Upsample block 1	Upsample + (Conv2d + BN)×2 + ReLU
Upsample block 2	Upsample + (Conv2d + BN)×2 + ReLU
Upsample block 3	Upsample + (Conv2d + BN)×2 + Sigmoid

Chapter 6

Experiments

6.1 Experiment Setting

Since the satellite image has a resolution of 0.229 meters per pixel, we define the length and width of the BEV space as 25.648 meters centered on vehicle position, equivalent to a size of 112×112 pixels on the aerial map. We also define the height of the BEV space as 2 meters. The space is divided into $28 \times 28 \times 5$ 3D grid cells, so that each cell represents a voxel of $0.916 \times 0.916 \times 0.4\, m^3$ in the real world. We utilize a temporal window of 5 seconds and randomly sample 5 frames in the window to compose a training sample.

We conduct two main experiments, one to compare against state-of-the-art VBL methods in GNSS-denied setting [24, 44], where we use 4 sequences and split them into 80% training, 20% testing data; and another to show our model's ability to generalize across different scenes given limited training data, where we use 2 sequences for training and 4 sequences for testing. The trajectory plots for sequences used in the cross-sequence testing experiment are shown in Fig. 6.1. In the figure, sequence 3 and 8 are used in training, sequence 4 to 7 are used in testing.

Training is distributed on 8 NVIDIA A100 GPUs for a total of 2500 epochs and with a learning rate of $4e^{-5}$. The configuration of the testing computer is described in Sec. 6.3.3 in system runtime.

During the testing phase, we crop and rotate the aerial map based on the GPS ground truth position as the center of the image with a size of 874×874 pixels, which

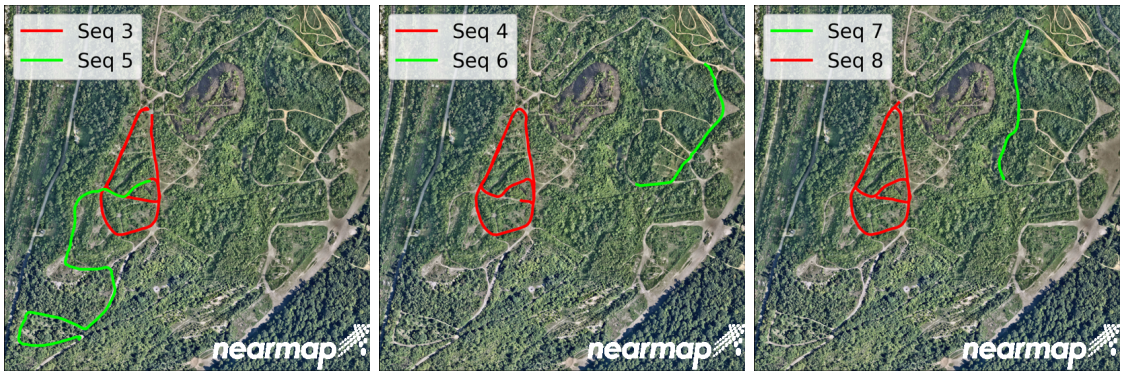

Figure 6.1: Trajectory plot for cross-sequence testing.

corresponds to a real-world coverage of approximately 200×200 square meters. This search region is sufficient to accommodate VIO drift for more than 10 minutes without a registration. For cross-sequence testing, we loosen the assumption of drifting range and use a search region of 100×100 square meters. Our camera system captures 3 frames per second and predicts registration consistent with camera frame; therefore, sufficient to prevent failure within the 100×100 square meter search range.

For template matching, NCC identifies the best match within the search area, maximizing similarity between the generated BEV image and the aerial map, thus predicting the vehicle's position relative to the aerial map. We observe failure cases where rendered BEV images are of moderate visual quality, where NCC fails in prediction. An example of failure cases is shown in Fig. 8.3.

6.2 Dataset Organization

We collect our real-world data set in the Pittsburgh area, with a VIO system on board. Detailed information on the sequences can be found in Table 6.1. For each training sample, we use the information of timestamp, trinocular RGB images, and GPS ground truth including x, y, and azimuth angle in the UTM coordinate system for training. The preprocessing process for cropping the aerial map can be found in Fig. 9.1.

Table 6.1: Statistics of GNSS-denied real-world dataset

	# images	traj. length (m)	coverage (m^2)
Seq 1	1634	1059.42	349.34×159.70
Seq 2	1563	1067.08	349.34×159.67
Seq 3	1427	1415.72	353.07×164.65
Seq 4	1210	1228.61	350.99×161.92
Seq 5	1707	1179.64	462.53×359.25
Seq 6	838	495.64	340.13×239.08
Seq 7	815	439.67	410.86×74.74
Seq 8	1395	1425.88	368.51×166.06
aerial map	-	-	1278.20×1646.46

6.3 Quantitative Comparison

We compare our method with GPS denied registration via occupancy mapping proposed in [24], and GeoDTR proposed in [44]. The comparison result is shown in Table 9.1.

Since GeoDTR is an image-retrieval-based method and relies on cultivating the corresponding information between camera inputs and polar transformation of aerial map images, it is required to preserve a database of candidate polar transformed images for real-world vehicle localization. We randomly sample 5000 particles within the search region at each timestamp and apply polar transformation according to the particle location on the map together with the azimuth angle of GPS ground truth. After obtaining the candidate polar images, for each timestamp, we pass in the camera images and polar images to the model, and calculate the distance between camera descriptors and polar descriptors, we choose candidate with closest descriptor distance as the top 1 prediction, and its corresponding real-world location as top 1 location, and we average the top 5 predicted locations as top 5 prediction.

6.3.1 Registration Accuracy

To evaluate the accuracy of vehicle registration, we calculate the mean and standard deviation (STD) of absolute position error (APE) between predicted position and the ground truth vehicle location provided by on-board GPS.

6.3.2 Registration Frequency

In the real-world localization scenario, the update frequency is another important factor that determines the stability of the registration system. We report the matching frequency by counting the total successful matches when the APE is within a threshold of 10 meters (the range we deem tolerable for our VIO system) and calculate the match rate as total successful matches divided by total camera frames for a sequence:

$$\mathbf{p}_i = (x_i, y_i), \tag{6.1}$$

$$\mathbf{d}^i_{\text{Euclidean}} = \|\mathbf{p}^i_{\text{gt}} - \mathbf{p}^i_{\text{pred}}\|_2, \tag{6.2}$$

$$p_{\text{match}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1} \cdot (\mathbf{d}^i_{\text{Euclidean}} < \mathbf{d}_{\text{threshold}}), \tag{6.3}$$

where N is the number of images for a sequence per camera module.

Remark 1 (Testing with Litman [24])**:** *It should be noted that the method proposed in [24] accumulates geometric features on a certain number of consecutive camera frames (50 by default), leading to a limited number of registration try-outs throughout a sequence. For comparison sake, we calculate the match rate as the total number of successful matches divided by the total number of occupancy maps synthesized in a sequence.*

Remark 2 (Testing with GeoDTR)**:** *It takes up to 21 hours to sample polar images for 5000 particles for 320 testing samples; therefore, we cannot further increase the density of particles. To apply image-retrieval-based method for on-board localization, it is required to have a pre-stored dataset, specifically in our case, of polar images sampled from all candidate positions on local aerial map enumerating all possible rotations, which is prohibitively expensive storage for on-board system in real-world localization.*

6.3.3 System Runtime

Testing is performed on a machine equipped with an AMD Ryzen 9 5900X 12-Core processor and a NVIDIA GeForce RTX 4090. The total time to localize 280 testing samples is 33.32 seconds, equivalent to 0.12 seconds to localize per camera frame. The camera frame rate for our system is 3 per second; therefore, our system is able to support online localization in a real world scenario.

6.4 Qualitative Comparison

Visualizations of the rendering and registration result can be found in Fig. 6.2. In this figure, the top row shows the rendering and registration result of our method, where the BEV images are highlighted in yellow boxes, the red dots indicate the NCC predictions from our system, and the blue dots indicate the GPS ground truth position. Our approach produces a coherent rendering to the aerial image. The bottom row shows the predictions from Litman [24]. Similarly, the red and blue dots indicate the predictions and ground truth, while the yellow boxes indicate the generated occupancy image overlayed on the ground truth. Only semi-dense rendering is available for Litman [24] (see the saturated white and green points around the red dots), resulting in compromised registration accuracy. The image rendering head

Figure 6.2: Qualitative comparison of our method and Litman [24]
Top row: predictions from our method, where the BEV images are highlighted in yellow boxes, the red dots indicate the NCC predictions, and the blue dots indicate ground truth positions. *Bottom row*: top-down view generated by Litman [24].

processes the encoded BEV feature with a spatial dimension of $64 \times 28 \times 28$ through a set of convolutional layers and 4 upsample layers, as shown in Table 5.1. The final BEV image is an RGB image with a size of 224×224 pixels, representing an

area of $51.296 \times 51.296\, m^2$. The occupancy map reconstructed from [24] aggregates geometric features from 50 consecutive frames, of which the coverage may vary for each prediction.

6.5 Model Generalization

To test the generalizability of the proposed system, we perform cross-sequence tests. Specifically, training with sequences 3 and 8 while testing with sequences 4-7. The trajectory plots for the sequences used in Fig. 6.1. The cross-sequence testing experiment is shown in in Table 6.2, we report search regions of $100 \times 100\, m^2$.

6.6 Ablation Study

In this section we explore the influence of choosing different hyperparameters and BEV space resolutions on the final registration result. Since the aerial map resolution is 0.229 meters, we experiment with the BEV grid resolutions of 0.458 meters and 0.916 meters, corresponding to 2 pixels and 4 pixels on the map, respectively. We also experiment with an increased number of layers and report the results in Table 6.3. Taking into account the result of the ablation study, we choose the resolution of the BEV grid as 0.916 meters, and the number of encoder layers as 1 for Table 9.1 and Table 6.2.

Table 6.2: Cross-sequence testing for model generalization

sequence	mean ↓	std ↓	match(%) ↑
Seq 4	11.24	6.64	45.38
Seq 5	13.77	6.74	31.16
Seq 6	12.72	6.38	36.63
Seq 7	16.30	6.92	21.81

Table 6.3: Ablation study on sequence 4

effects of architecture choice and hyper parameters

# layers	grid reso. (m)	# params	mean ↓	std ↓	match(%) ↑
1	0.458	1.71M	27.47	27.83	48.75
2	0.458	2.09M	27.75	27.32	45.42
1	0.916	1.44M	21.17	25.49	57.50
2	0.916	1.72M	36.40	25.66	20.42

Chapter 7

Integration with Visual Inertial Odometry

7.1 Registration Factor Graph Formulation

To integrate the map registration module with a VIO system, a factor graph is constructed and optimized, as in Fig. 7.1. Since VIO produces estimation at the camera frame rate, and the map registration module aggregates consecutive frames within a temporal window, as well as produces a registration result every n seconds, which is less frequent comparex to the VIO module. Instead of integrating all sensor measurements into one factor graph, we split VIO and map registration into two separate factor graphs.

The VIO graph handles measuremnts from IMU, stereo visual odometry at a higher frequency; the registration graph takes in optimized VIO poses and adds them as between factors; the registration graph simplifies to Fig. 7.1.

7.2 Odometry Integration Test on Real-world Robot

To demonstrate the performance of the map registration module in the state estimation system, we train BEVRender with *Seq 4*, and perform state estimation on another log,

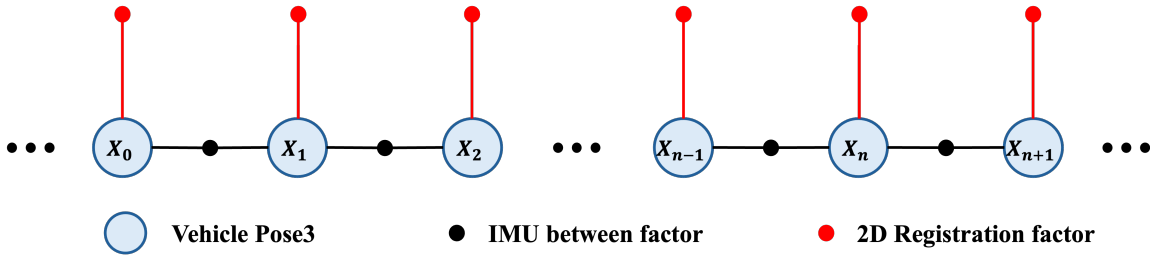

Figure 7.1: Simplified factor graph for map registration

we report the APE metric between VIO only and the registration integrated result in Fig. 7.3. The corresponding statistics and histogram are shown in Fig. 7.2 and Fig. 7.4. We also report a visualization of VIO and registration integrated trajectories

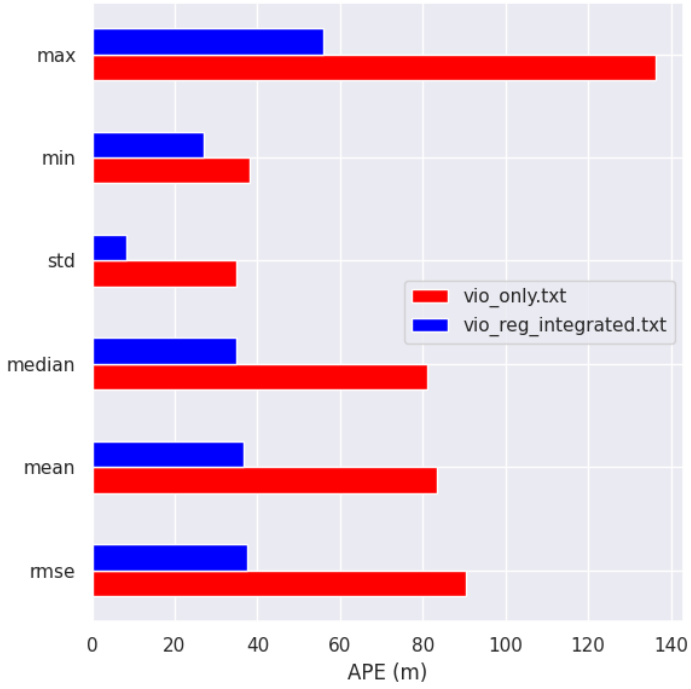

Figure 7.2: Statistics of APE on VIO and registration integrated VIO

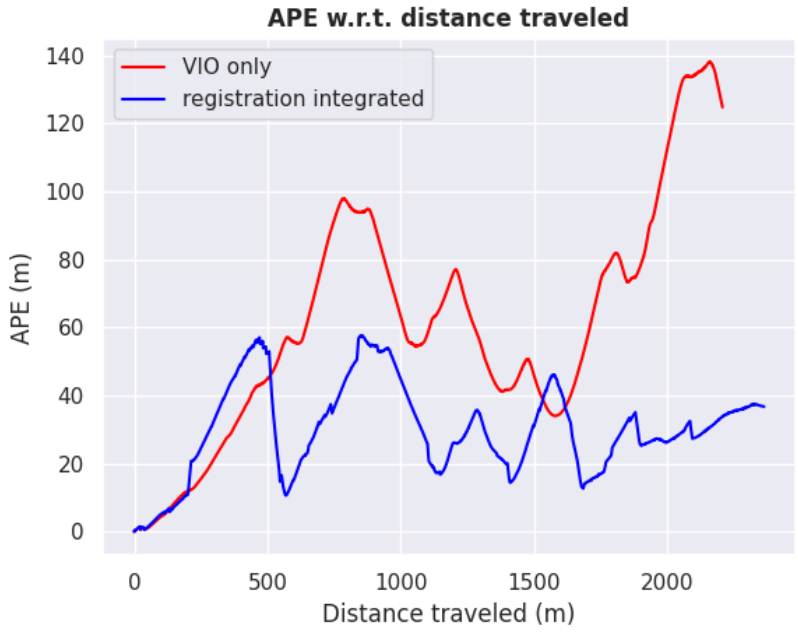

Figure 7.3: APE on VIO and registration integrated VIO

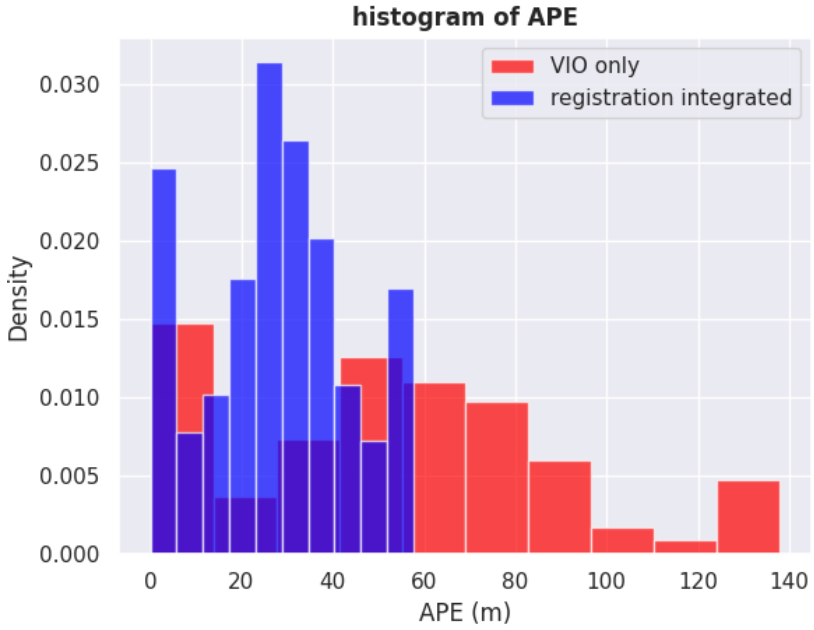

Figure 7.4: Histogram of APE on VIO and registration integrated VIO

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this work, we present a learning-based system to generate local BEV images combined with NCC for ground vehicle localization in GNSS-denied off-road environments. Our system incorporates the deformable attention module with BEVFormer for a multiview camera sensor setting, followed by a novel rendering head to generate high-precision BEV images to enable downstream localization task. Through experiments with real-world data, we show that BEVRender, despite its lightweight structure, is capable of learning local BEV representation and effectively reduce VIO drift when integrated into state estimation system.

8.2 Future Work

8.2.1 Cross-season Map Registration

To enhance our ground vehicle localization system for operation across different seasons, future research will focus on improving the ability of the network to learn and generalize features from diverse seasonal landscapes. This is essential for deploying our system in real-world scenarios where environmental conditions fluctuate significantly over the year. Furthermore, our goal is to advance the fidelity of BEV image generation by incorporating techniques such as the diffusion module, inspired by the diffusion

transformer [30]. This enhancement is expected to refine the detail and precision of the BEV images, thus enriching contextual data for more accurate vehicle localizations.

8.2.2 Epipolar Transformer for Improved Feature Encoding

BEVRender extracts token from ground camera image and aggregates visual information from consecutive frames to construct BEV representation. In essence, our approach encodes 2D visual information given VIO pose prediction and lacks the ability to encode 3D geometric clues in the model.

To effectively learn geometric information in the BEV space from ground camera images, an alternative way is to utilize stereo information and find ways to encode the epipolar constraint (see Fig. 8.1) in model propagation as in the Epipolar Transformer [14]. After constructing a 3D volume for a single frame, aggregating volumes over consecutive frames to reference local BEV space to utilize 3D clues in the model, and further using local aerial map as supervision for BEV generation, as shown in Fig. 8.2,

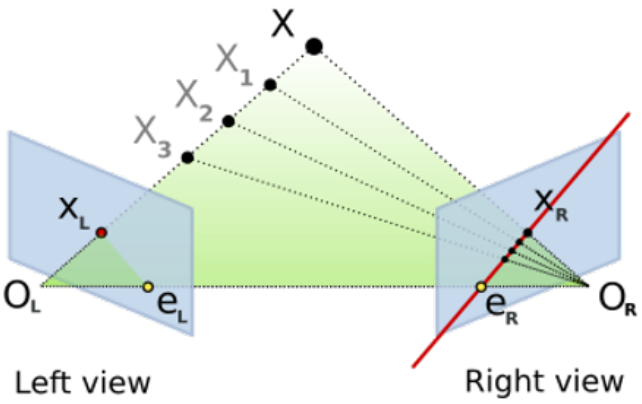

Figure 8.1: Epipolar geometry

8.2.3 Other Future Directions

Further improvements will also explore the integration of temporal features to accumulate historical data more effectively, addressing current limitations caused by

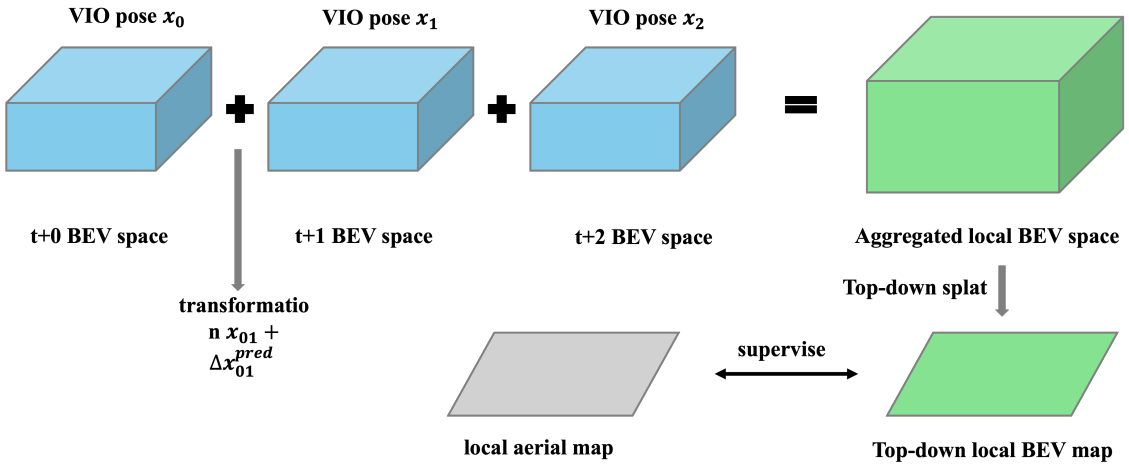

Figure 8.2: Aggregation of Volume

projection adjustments and changes in vehicle pose. Moreover, explorations can be made on removing dependence on GPS information for training by leveraging local state estimates from VIO.

In addition, a transition from classic template matching to learnable template matching for vehicle positioning is anticipated to overcome the limitation of NCC's uniform pixel weighting, as shown in Fig. 8.3, and to enable the system to prioritize strategically significant areas, potentially elevating the accuracy of vehicle registration in challenging environments.

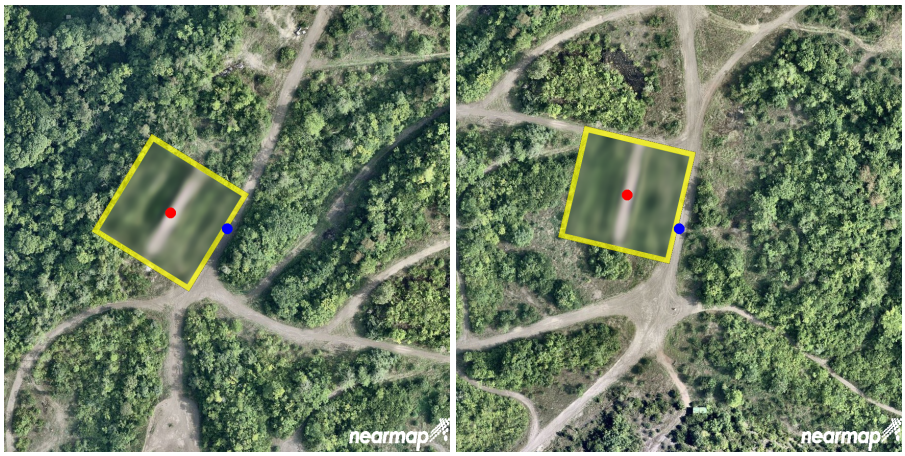

Figure 8.3: Examples of failure cases due to the uniform weighting of NCC

Chapter 9

Appendix: Supplement of Figures and Tables

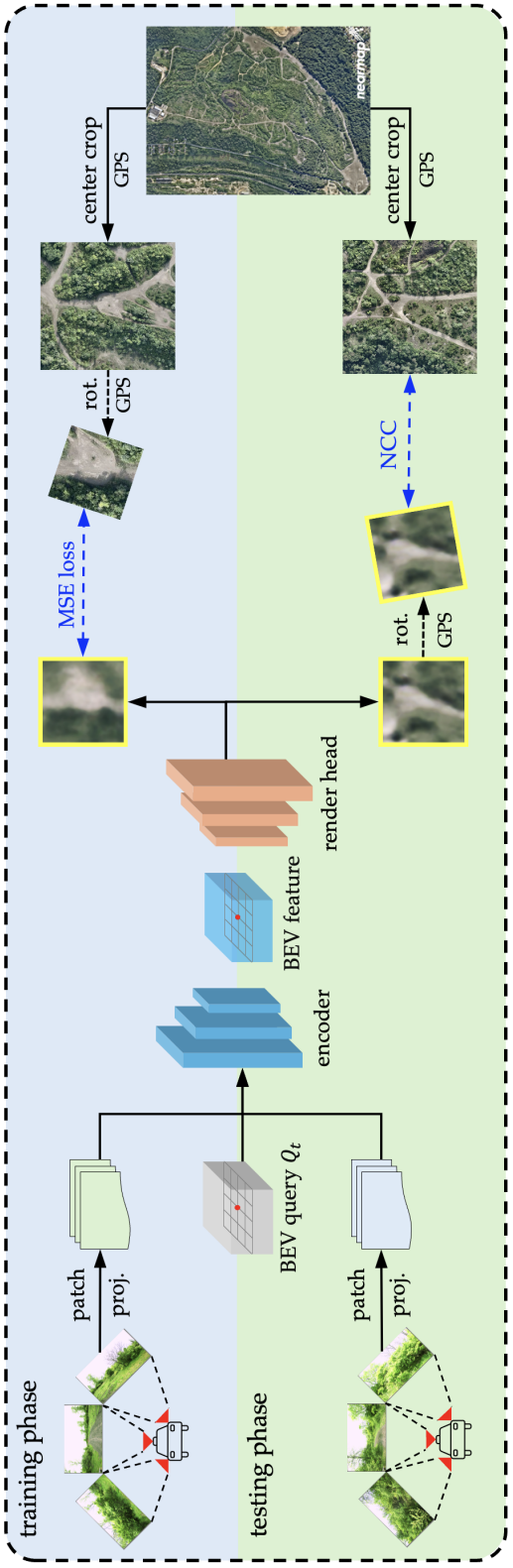

Figure 9.1: System diagram
The light blue background indicates the training phase and the light green background indicates the testing phase. During the training phase, camera images are patch projected and sent to the feature encoder (in blue) and rendering head (in orange) to generate BEV images (highlighted in yellow boxes). The aerial map image is rotated and cropped according to the GPS information provided, ensuring that the final label image accurately represents the BEV space surrounding the vehicle. During the testing phase, the rendered BEV image is rotated according to the azimuth angle provided by the GPS, and matched against a local search region surrounding the vehicle position.

Table 9.1: Quantitative comparison with real-world dataset

approach	*Seq 1*			*Seq 2*			*Seq 3*			*Seq 4*		
	mean ↓	std ↓	match (%) ↑	mean ↓	std ↓	match (%) ↑	mean ↓	std ↓	match (%) ↑	mean ↓	std ↓	match (%) ↑
Litman [24]	24.35	13.50	21.62 (Rmk.1)	34.45	21.59	12.12	26.27	13.44	11.46	61.04	55.80	8.89
GeoDTR [44] (top 1)	82.72	25.52	0.00 (Rmk.2)	90.27	29.92	1.28	84.40	27.33	0.36	86.53	27.60	0.00
GeoDTR [44] (top 5 avg.)	67.35	24.22	0.94	74.06	30.43	1.60	71.33	28.91	1.07	66.34	29.35	1.67
Ours	19.33	26.09	63.44	22.40	27.92	60.90	20.60	24.96	58.93	21.18	25.49	57.50

1. The darker shading indicates the best results, and the lighter shading indicates the second-best results.
2. The mean and std are calculated for the APE for predicted positions, see the registration accuracy of Sec. 6.3.1 for more details.
3. The search region is set to 200×200 square meters.

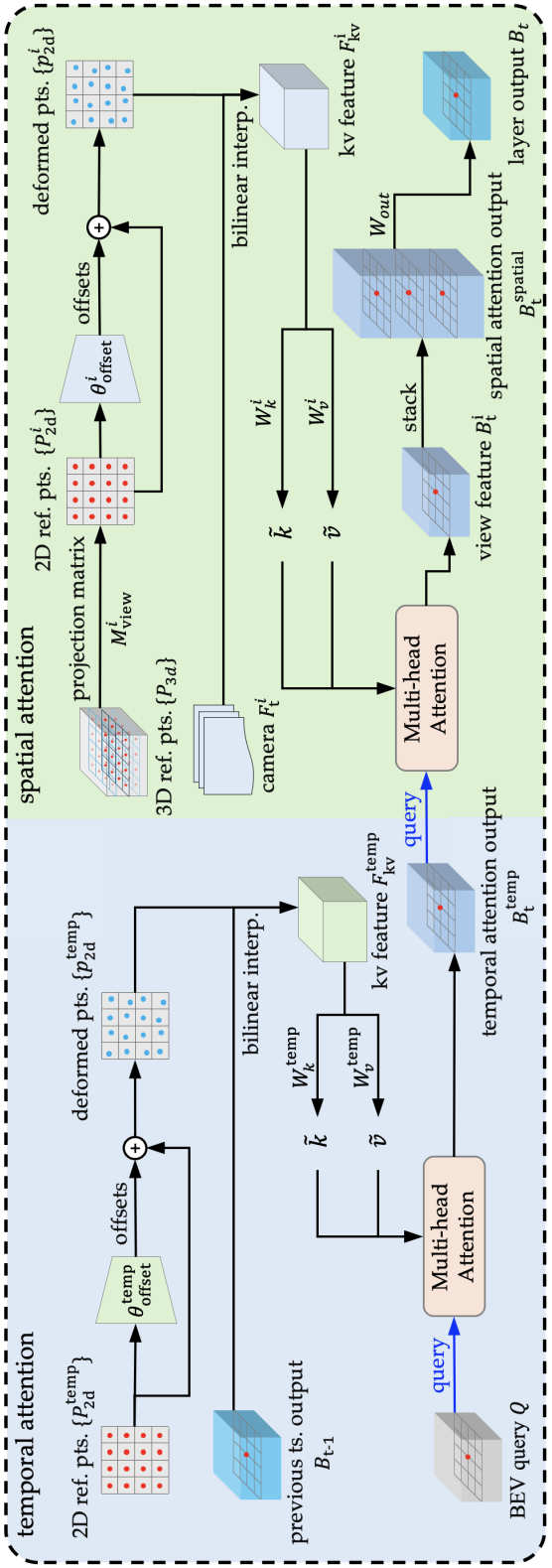

Figure 9.2: Encoder layer architecture

An encoder layer is composed of temporal and spatial attention. In temporal attention, a set of 2D reference points with a spatial dimension of $l \times w$ is sampled and deformed. Next, bilinear sampling is performed to extract tokens for multi-head attention (MHA) [39] given deformed reference points from previous timestamp BEV feature B_{t-1}. The MHA output from temporal attention serves as a query for the subsequent spatial attention module. In spatial attention, we sample one point per 3D grid cell in the BEV space as reference points and project them to the three camera image frames with extrinsic and intrinsic parameters to obtain 2D reference points for each image view. Similarly to temporal attention, the 2D reference points are deformed and used for bilinear sampling, but from camera feature. A more detailed description can be found in Sec. 5.1.

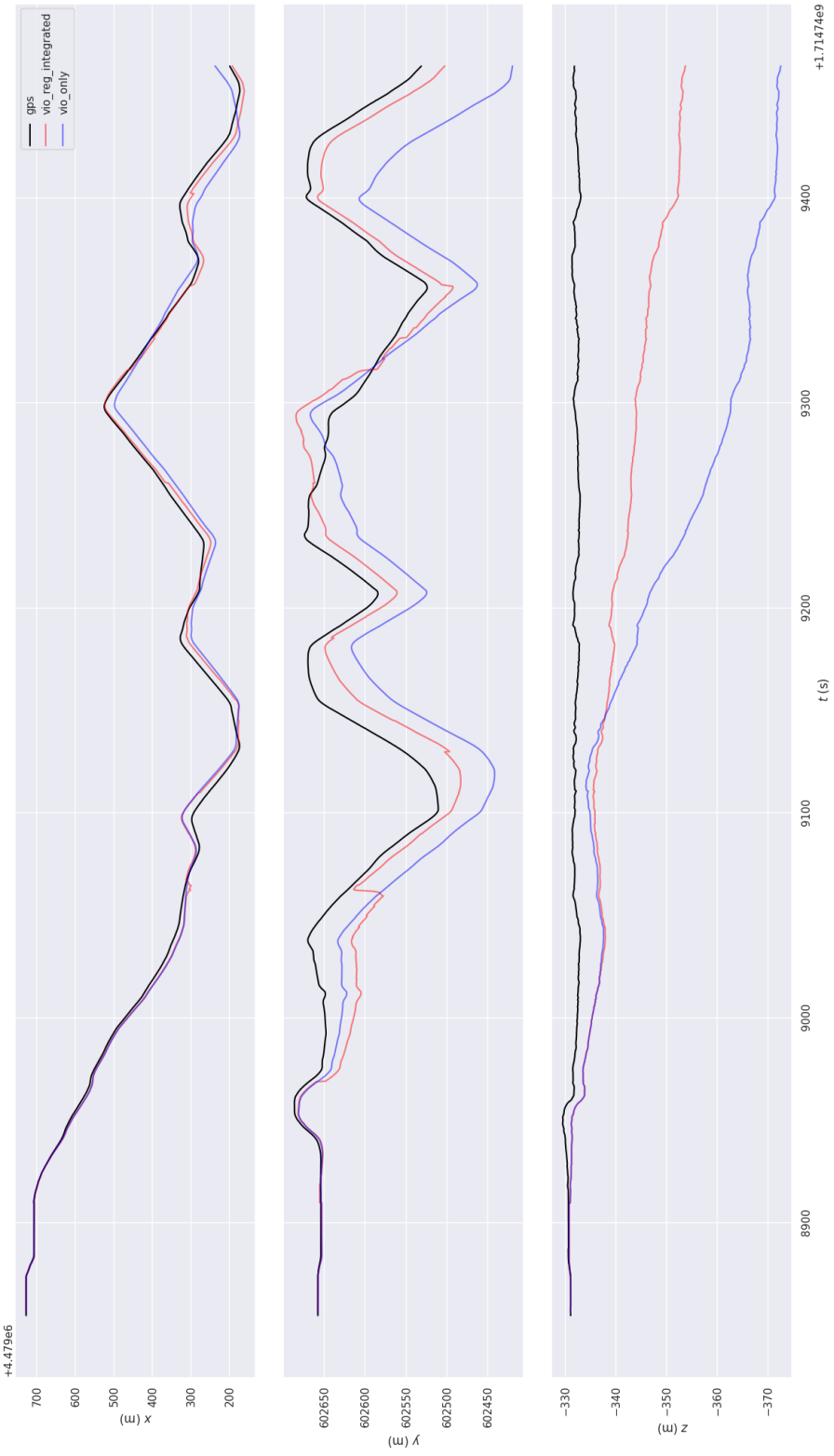

Figure 9.3: Comparison between VIO and registration integrated trajectory in x, y and z

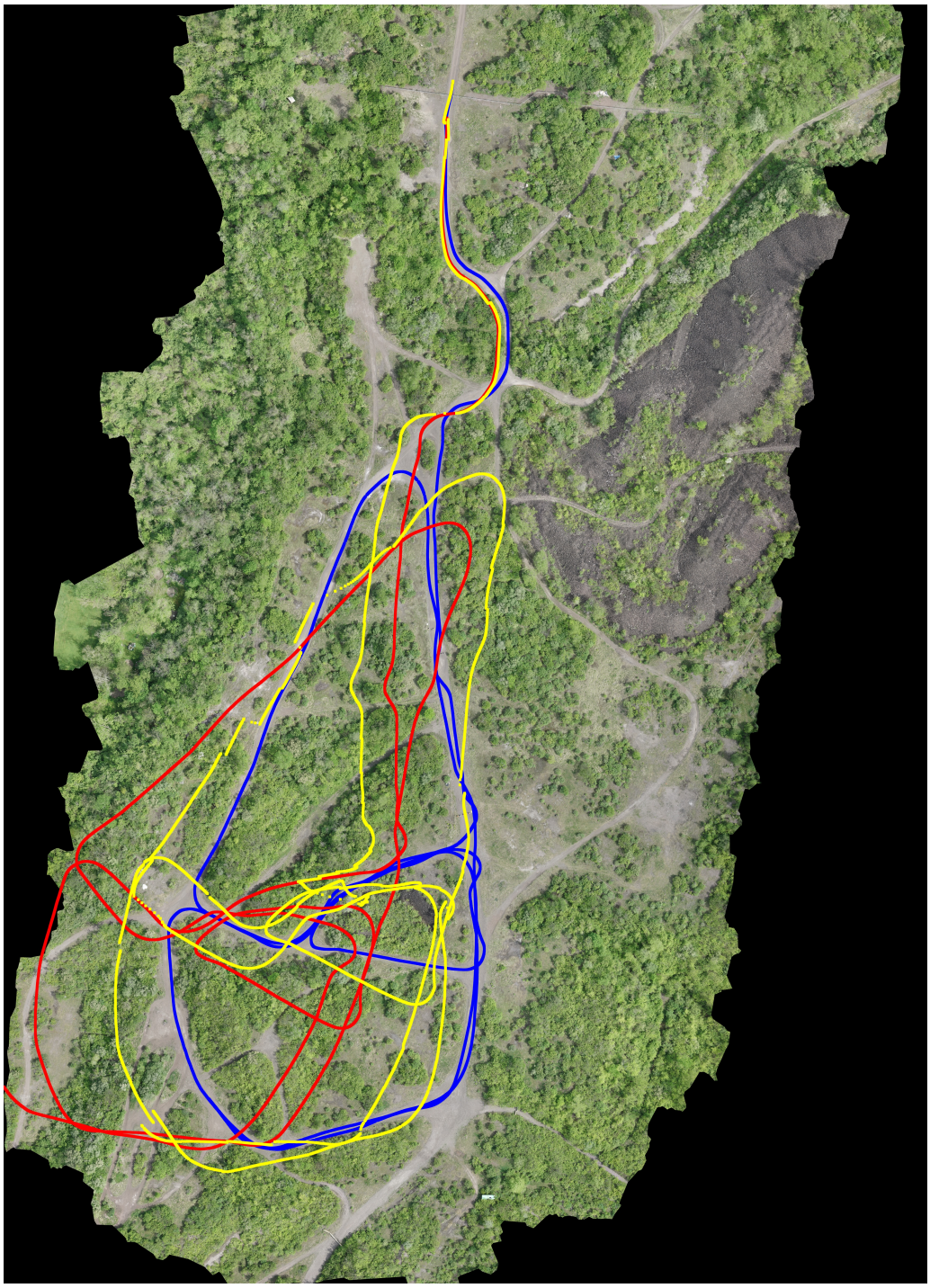

Figure 9.4: Visualization of VIO trajectory and registration integrated trajectory red - VIO, blue - GPS, yellow - registration integrated trajectory

Bibliography

[1] Adil Kaan Akan and Fatma Güney. Stretchbev: Stretching future instance prediction spatially and temporally. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 444–460, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19839-7. 3.3

[2] Yusra Alkendi, Lakmal Seneviratne, and Yahya Zweiri. State of the art in vision-based localization techniques for autonomous navigation systems. *IEEE Access*, 9:76847–76874, 2021. doi: 10.1109/ACCESS.2021.3082778. 1.1

[3] Hongxiang Cai, Zeyuan Zhang, Zhenyu Zhou, Ziyin Li, Wenbo Ding, and Jiuhua Zhao. Bevfusion4d: Learning lidar-camera fusion under bird's-eye-view via cross-modality guidance and temporal aggregation, 2023. 3.3

[4] Andrea Boscolo Camiletto, Alfredo Bochicchio, Alexander Liniger, Dengxin Dai, and Abel Gawel. U-bev: Height-aware bird's-eye-view segmentation and neural map-based relocalization, 2023. 3.2

[5] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proc. IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 15661–15670, October 2021. 3.3

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 9650–9660, October 2021. 3.2

[7] Chang Chen, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Trans4map: Revisiting holistic bird's-eye-view mapping from egocentric images to allocentric semantics with vision transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4013–4022, January 2023. 3.3

[8] Ziye Chen, Kate Smith-Miles, Bo Du, Guoqi Qian, and Mingming Gong. An efficient transformer for simultaneous learning of bev and lane representations in

3d lane detection, 2023. URL https://arxiv.org/abs/2306.04927. 3.3

[9] F. Dellaert and M. Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics (FNT)*, 6(1-2):1–139, 2017. doi: 10.1561/2300000043. 3.1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Intl. Conf. on Learning Representations (ICLR)*, 2021. 1.3, 3.1

[11] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. C-bev: Contrastive bird's eye view training for cross-view image retrieval and 3-dof pose estimation, 2023. 3.2

[12] James Gunn, Zygmunt Lenyk, Anuj Sharma, Andrea Donati, Alexandru Buburuzan, John Redford, and Romain Mueller. Lift-attend-splat: Bird's-eye-view camera-lidar fusion using transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4526–4536, June 2024. 3.3

[13] Yao He, Ivan Cisneros, Nikhil Keetha, Jay Patrikar, Zelin Ye, Ian Higgins, Yaoyu Hu, Parv Kapoor, and Sebastian Scherer. Foundloc: Vision-based onboard aerial localization in the wild. *arXiv preprint arXiv:2310.16299*, 2023. 3.2

[14] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8.2.2

[15] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016. 3.1

[16] Chunyong Hu, Hang Zheng, Kun Li, Jianyun Xu, Weibo Mao, Maochun Luo, Lingxuan Wang, Mingxia Chen, Qihao Peng, Kaixuan Liu, Yiru Zhao, Peihan Hao, Minzhe Liu, and Kaicheng Yu. Fusionformer: A multi-sensory fusion in bird's-eye-view and temporal consistent transformer for 3d object detection, 2023. 3.3

[17] Qi Jiang and Hao Sun. Semanticbevfusion: Rethinking lidar-camera fusion in unified bird's-eye view representation for 3d object detection. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5707–5714, 2023. doi: 10.1109/IROS55552.2023.10342368. 3.3

[18] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition, 2023. 3.2

[19] Gowtham Kuppudurai, Kyu-young Hwang, Hyeon-Gyu Park, and Youngwook Kim. Localization of airborne platform using digital elevation model with adaptive weighting inspired by information theory. *IEEE Sensors Journal*, 18 (18):7585–7592, 2018. doi: 10.1109/JSEN.2018.2859396. 3.1

[20] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, Hao Tian, Enze Xie, Jiangwei Xie, Li Chen, Tianyu Li, Yang Li, Yulu Gao, Xiaosong Jia, Si Liu, Jianping Shi, Dahua Lin, and Yu Qiao. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Trans. Pattern Anal. Machine Intell.*, 46 (4):2151–2170, 2024. doi: 10.1109/TPAMI.2023.3333838. 3.3

[21] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):1477–1485, Jun. 2023. doi: 10.1609/aaai.v37i2.25233. URL https://ojs.aaai.org/index.php/AAAI/article/view/25233. 3.3

[22] 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 *European Conf. on Computer Vision (ECCV)*, pages 1–18. Springer, 2022. 1.3, 1.4, 3.3, 5.1

[23] Pan Liao, Feng Yang, Di Wu, and Liu Bo. Monodetrnext: Next-generation accurate and efficient monocular 3d object detection method, 2024. URL https://arxiv.org/abs/2405.15176. 3.3

[24] Yehonathan Litman, Daniel McGann, Eric Dexheimer, and Michael Kaess. Gps-denied global visual-inertial ground vehicle state estimation via image registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8178–8184, 2022. doi: 10.1109/ICRA46639.2022.9812364. (document), 1.3, 3.1, 6.1, 6.3, 1, 6.4, 6.2, 6.4, 9.1

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021. 5.2

[26] Abdelhak Loukkal, Yves Grandvalet, Tom Drummond, and You Li. Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 51–60, January 2021. 3.3

[27] Sambit Mohapatra, Senthil Yogamani, Varun Ravi Kumar, Stefan Milz, Heinrich Gotzig, and Patrick Mäder. Lidar-bevmtn: Real-time lidar bird's-eye view multi-task perception network for autonomous driving, 2023. URL https:

//arxiv.org/abs/2307.08850. 3.3

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 3.2

[29] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. doi: 10.1109/LRA.2020.3004325. 3.3

[30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 4195–4205, 2023. 8.2.1

[31] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5935–5943, January 2023. 3.3

[32] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *IEEE Intl. Conf. on intelligent transportation systems (ITSC)*, pages 1672–1679. IEEE, 2018. 3.1

[33] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8690–8699, October 2023. 3.3

[34] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5133–5139, 2021. doi: 10.1109/ICRA48506.2021.9561169. 3.3

[35] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9200–9206, 2022. doi: 10.1109/ICRA46639.2022.9811901. 3.3

[36] 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 *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2023. 3.1, 3.2

[37] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. SNAP: Self-Supervised Neural Maps for Visual Positioning and Semantic Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1.1, 1.3, 3.2

[38] Siddharth Srivastava, Frederic Jurie, and Gaurav Sharma. Learning 2d to 3d lifting for object detection in 3d for autonomous vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4504–4511, 2019. doi: 10.1109/IROS40897.2019.8967624. 3.3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 3.3, 5.2, 9.2

[40] Anirudh Viswanathan, Bernardo R. Pires, and Daniel Huber. Vision based robot localization by ground to satellite matching in gps-denied situations. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 192–198, 2014. doi: 10.1109/IROS.2014.6942560. 3.1

[41] Xue Wan, Yuanbin Shao, Shengyang Zhang, and Shengyang Li. Terrain aided planetary uav localization based on geo-referencing. *IEEE Trans. on Geoscience and Remote Sensing*, 60:1–18, 2022. 3.1

[42] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *IEEE Trans. Robotics*, 21(2):208–216, 2005. doi: 10.1109/TRO.2004.835453. 1.3, 3.2

[43] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 4794–4803, June 2022. 1.4, 3.3, 5.2, 5.2

[44] Xiaohan Zhang, Xingyu Li, Waqas Sultani, Yi Zhou, and Safwan Wshah. Cross-view geo-localization via learning disentangled geometric layout correspondence. In *AAAI Conf. on Artificial Intelligence*, volume 37, pages 3480–3488, 2023. 1.1, 1.3, 3.2, 6.1, 6.3, 9.1

[45] Zhihuang Zhang, Meng Xu, Wenqiang Zhou, Tao Peng, Liang Li, and Stefan Poslad. Bev-locator: An end-to-end visual semantic localization network using multi-view images, 2022. 3.2

[46] Junyu Zhu, Lina Liu, Yu Tang, Feng Wen, Wanlong Li, and Yong Liu. Semi-supervised learning for visual bird's eye view semantic segmentation, 2024. URL `https://arxiv.org/abs/2308.14525`. 3.3

[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection.

arXiv preprint arXiv:2010.04159, 2020. 5.2