Supplementary Material for Fine-Grained Cross-View Geo-Localization Using a Correlation-Aware Homography Estimator

In this **supplementary document**, we provide detailed information on projecting ground images to bird's-eye view for VIGOR [4] and KITTI [2] datasets in Section A, as well as the computation details for converting GPS labels to pixel coordinates used during training in Section B. We also discuss the potential for future research based on our method in Section C.

A Projection Details

A.1 Details for Spherical Transform

In the main paper, we provide an overview of the derivation process of the Spherical Transform and the final result in Equation 1. Here we will provide detailed information on the process for the projection. We will use the expressions defined in Section 3.1. To clarify the presentation, we also provide the schematic diagram of the projection process, as shown in Figure 6.

In the camera coordinates, the conversion formula between the Cartesian coordinates $P = (x_1, y_1, z_1)$ and the spherical coordinates (ϕ, θ) is given by:

$$\begin{cases} \phi = \text{arctan2}\left(y_1, x_1\right) & \in [-\pi, \pi], \\ \theta = \text{arctan2}\left(z_1, \sqrt{x_1^2 + y_1^2}\right) & \in [-\pi/2, \pi/2]. \end{cases} \tag{6}$$

The equirectangular projection is used to project spherical coordinates onto a plane, which is the display format for panoramic images. The conversion formula between the spherical coordinates (ϕ, θ) and the Normalised Equirectangular coordinates $P' = (x_2, y_2)$ is given by:

$$\begin{cases} x_2 = \dfrac{-\phi}{\pi} & \in [-1, 1], \\ y_2 = \dfrac{\theta}{\pi/2} & \in [-1, 1]. \end{cases} \tag{7}$$

The negative sign in the x_2 expression is due to the fact that panoramic images display the scene as viewed from the camera's optical center to the outside. Therefore, when ϕ is positive, it corresponds to the negative half-plane of the Equirectangular plane in terms of x_2. The mapping between pixel coordinates (u_p, v_p) and Normalised Equirectangular coordinates (x_2, y_2) on the panorama is established as:

$$\begin{cases} u_p = (x_2 + 1)W_p/2 & \in [0, W_p], \\ v_p = (-y_2 + 1)H_p/2 & \in [0, H_p]. \end{cases} \tag{8}$$

To obtain the corresponding bird's-eye view of the panorama, we place a tangent plane at the south pole of the spherical imaging plane as a new imaging plane, as shown in Figure 6 (c). The focal length of the BEV in the imaging process is $f = 0.5W_b / \tan(fov)$. The camera coordinate system coordinates (x_1, y_1, z_1) corresponding to a pixel (u_b, v_b) on the bird's-eye view (BEV) imaging plane are determined as:

$$\begin{cases} x_1 = -v_b + H_b/2, \\ y_1 = -u_b + W_b/2, \\ z_1 = -f. \end{cases} \tag{9}$$

By substituting Equation 9 into Equation 6, and then substituting the result into Equation 7, we can obtain the Normalised Equirectangular coordinates (x_2, y_2) for a pixel on the bird's-eye view (BEV) image plane. Finally, by substituting the Normalised Equirectangular coordinates into Equation 8, we can obtain the pixel coordinates (u_p, v_p) on the panoramic image. This leads to the Spherical Transform mapping formula in Equation 1.

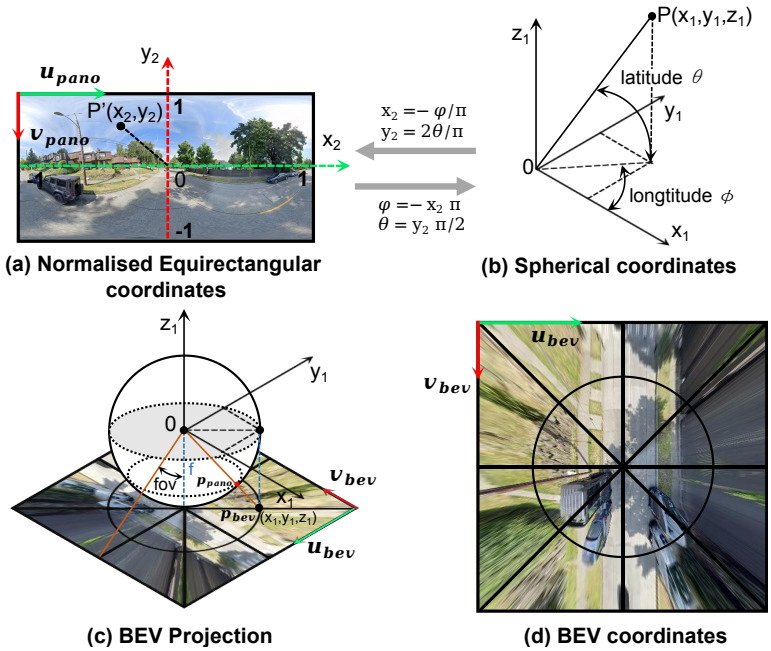


Figure 6: Illustration of panorama imaging model and the spherical transform mechanism for projecting panoramas to a bird's-eye view.

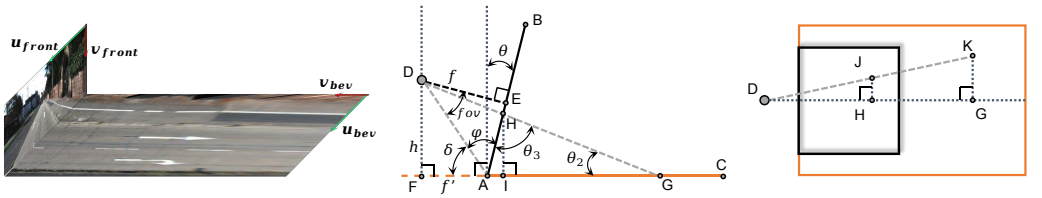
Figure 7: The examples of bird's-eye view images obtained through the Spherical Transform at different field of view (fov) parameters.

We test the effectiveness of the bird's-eye view projection with different field of view (fov) parameters, as shown in Figure 7. As seen in (d), when $fov = 85°$, the field of view of the bird's-eye view image is similar to that of the corresponding satellite image, and there are few invalid parts in the image. Therefore, we use this parameter for all of our experiments.

A.2 Details for Projecting Front View Images in KITTI

In the KITTI dataset [2], the ground camera captures the front view image, so the process of projecting it onto a bird's-eye view is completely different from that of the panorama in VIGOR [4]. We illustrate the mechanism of this projection process in Figure 8, where AB represents the height of the front view image, and AC represents the desired height of the bird's-eye view image.

We define the size of the front view image as $H_f \times W_f$, and the target size of the bird's-eye view image as $H_b \times W_b$. Using a method similar to that described in Section 3.1, we imagine a bird's-eye projection plane parallel to the ground and adjacent to the front view image. Then we connect the



(a) Front-View and BEV Image (b) Front View of Projection (c) Top View of Projection

Figure 8: Illustration of the projection mechanism for KITTI [2].

camera's optical center with each pixel on the bird's-eye view image to obtain the corresponding pixel coordinates on the front view image.

In Figure 8 (b), D represents the camera's optical center, fov represents the field of view of the front view image ($\angle ADE$), f represents the focal length DE, h represents the distance from the camera's optical center D to the bird's-eye projection plane, and θ represents the angle between the camera's imaging plane AB and the vertical direction. To facilitate the subsequent derivation, we also define δ to represent $\angle FAD$, φ to represent $\angle BAD$, θ_2 to represent $\angle AGD$, θ_3 to represent $\angle AHG$, f' to represent FA, and l_0 to represent AD. According to [2], $fov = 17.5°$ in the KITTI dataset. These variables are calculated as follows:

$$\begin{cases} f = \dfrac{H_f}{2} / \tan(fov) \\ \varphi = \dfrac{\pi}{2} - fov \\ \delta = \dfrac{\pi}{2} - (\varphi - \theta) \\ l_0 = \sqrt{f^2 + \left(\dfrac{H_f}{2}\right)^2} \\ h = l_0 \sin\delta \\ f' = l_0 \cos\delta. \end{cases} \tag{10}$$

We denote the pixel coordinates on the bird's-eye view imaging plane as (u_b, v_b). The values of θ_2 and θ_3 can be calculated using the arctangent formula and the properties of exterior angle as follows:

$$\begin{cases} \theta_2 = \arctan\left(h / \left(f' + H_b - v_b\right)\right) \\ \theta_3 = \dfrac{\pi}{2} + \theta - \theta_2. \end{cases} \tag{11}$$

Then, based on the low of sines and similarity triangle properties, we can calculate the corresponding pixel coordinates (u_f, v_f) on the front view image for each pixel on the bird's-eye view image. The calculation is as follows:

$$\begin{cases} \dfrac{AH}{AG} = \dfrac{\sin\theta_2}{\sin\theta_3} \quad \Rightarrow \dfrac{H_f - v_f}{H_b - v_b} = \dfrac{\sin\theta_2}{\sin\theta_3} \\ \dfrac{JH}{KG} = \dfrac{DH}{DG} = \dfrac{FI}{FG} \quad \Rightarrow \dfrac{W_f/2 - u_f}{W_b/2 - u_b} = \dfrac{f' + (H_f - v_f)\sin\theta}{f' + H_b - v_b} \end{cases} \tag{12}$$

$$\Rightarrow \begin{cases} v_f = H_f - \dfrac{\sin\theta_2}{\sin\theta_3}(H_b - v_b) \\ u_f = \dfrac{W_f}{2} - \dfrac{f' + (H_f - v_f)\sin\theta}{f' + H_b - v_b}\left(\dfrac{W_b}{2} - u_b\right) \end{cases} \tag{13}$$

We introduce the angle θ because we found that in practical testing, the imaging plane of the ground camera may not be perfectly vertical, as shown in Figure 9. Based on experimental results, we use $\theta = 0.8°$ for all experiments. During the actual projection process, we set the target size of the bird's-eye view image to be $H_b \times W_b = (6 * W_f) \times (6 * W_f)$, which corresponds to a field of view of approximately $40m \times 40m$. After the first projection calculation, we directly applied the homography

(a) Front-View Image (b) Satellite Image

(c) θ = 0° (d) θ = 0.8° (e) θ = 1.6° (f) θ = 0.8° and Rotate

Figure 9: The examples of bird's-eye view images obtained from front view image at different θ parameters.

matrix H obtained from the above projection process to obtain the BEV. Then, we used scaling and rotation homography matrices H_1 and H_2 to obtain the final BEV with the specified size.

$$H_1 = \begin{bmatrix} \text{scale} & 0 & 0 \\ 0 & \text{scale} & 0 \\ 0 & 0 & 1 \end{bmatrix}, H_2 = \begin{bmatrix} \cos\gamma & -\sin\gamma & u_c(1-\cos\gamma) + v_c\sin\gamma \\ \sin\gamma & \cos\gamma & v_c(1-\cos\gamma) - u_c\sin\gamma \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Here, scale represents the ratio of the current $H_b \times W_b$ to the desired input size of the network (512×512). γ represents the yaw angle of the ground camera with noise, and (u_c, v_c) represent the pixel coordinates of the center of the final BEV image.

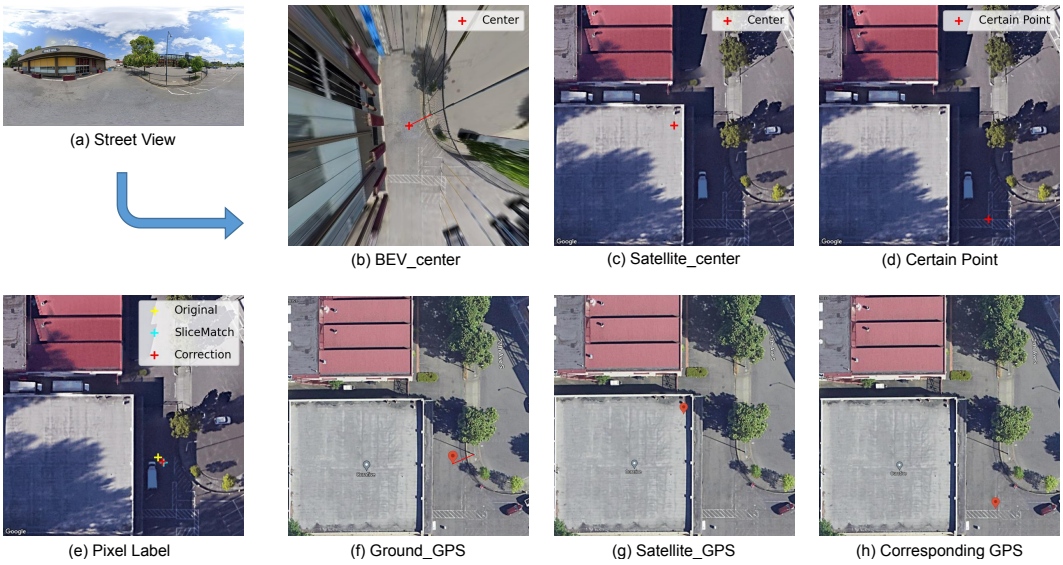
B Label Correction

B.1 Label Correction in VIGOR Dataset

During research, we found that the ground truth for pixel coordinates of ground images on satellite patches in the VIGOR dataset [4] is inaccurate. The VIGOR dataset uses a uniform meter-to-pixel resolution for converting the latitude and longitude of ground images to their location in aerial images across all four cities. SliceMatch [3] also discovered this issue and proposed a correction, but their improvement is limited to using different average meter-to-pixel resolutions for each city, which may still result in some inaccuracies in different regions of the same city.

We propose the use of Mercator projection [1] to directly compute the pixel coordinates of ground images on specified satellite images using the GPS information provided in the dataset. Equation 4 allows us to calculate the pixel coordinates of a GPS coordinate on a global scale. We compute the pixel coordinates of the satellite patch's GPS label and the pixel coordinates of the ground image's GPS label. We then add the difference between them to the center coordinates of the current satellite patch, resulting in accurate pixel coordinates of the ground image on the satellite patch. The result of the correction is shown in Figure 10, which shows that the label provided by VIGOR has a significant deviation from the true position. The label corrected by SliceMatch [3] reduces this deviation, but there is still a small offset between the corrected label and the true position. The statistical information on the absolute error, measured in meters, between the labels provided by VIGOR [4] and the corrected labels using our method, as well as the absolute error between the labels provided by SliceMatch [3] and the corrected labels using our method is presented in Table 5.

Note that in the model training process, including our method and all other methods, we do not directly use the GPS coordinates for training. This is because if GPS is used directly to calculate the



(a) Street View (b) BEV_center (c) Satellite_center (d) Certain Point

(e) Pixel Label (f) Ground_GPS (g) Satellite_GPS (h) Corresponding GPS

Figure 10: Illustration of label problem in VIGOR dataset [4]. In (e), we show the label provided by the VIGOR [4], the label corrected by SliceMatch [3], and the pixel coordinates computed using our method. The space distance between label provided by VIGOR and ours reaches 1.93 m. The pair of (d) and (h) represent the the pixel coordinates obtained by our method and their corresponding GPS location on Google Maps, demonstrating the accuracy of our method.

Table 5: Absolute error statistics for labels in VIGOR [4] and SliceMatch [3] in four cities. The absolute error is defined as the distance between the original and the corrected locations.

City	VIGOR (m)				SliceMatch (m)			
	Min.	Mean	Median	Max.	Min.	Mean	Median	Max.
Chicago	0.00	0.44	0.45	0.80	0.00	0.10	0.13	0.31
New York	0.00	0.20	0.21	0.39	0.00	0.16	0.16	0.42
San Francisco	0.00	0.42	0.45	0.86	0.00	0.09	0.13	0.27
Seattle	0.00	1.73	1.80	3.13	0.00	0.12	0.13	0.33

training loss, the data truncation problem will occur when using Float32 tensor format since the valid data is five decimal places, and trigonometric functions are used in the calculation.

B.2 Label Creation in KITTI Dataset

Supervised information required for training our network includes the pixel location on the bird's-eye view image, as well as its corresponding true GPS value. However, obtaining this information directly from the KITTI dataset [2] is not feasible, and we need to calculate it ourselves.

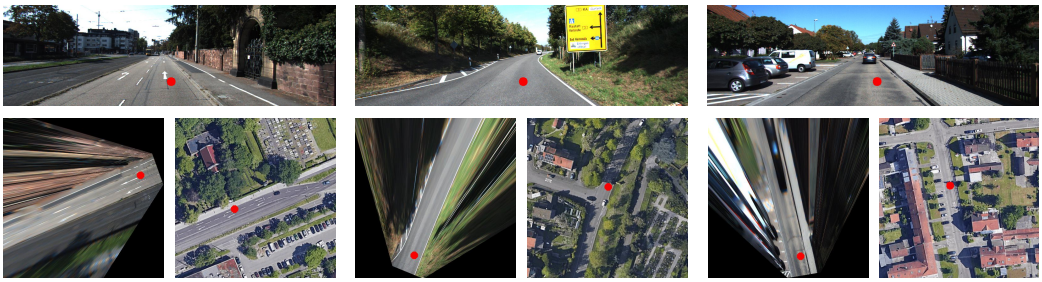
To obtain the required supervised information, we use the calibration files provided by KITTI and the point cloud data in the dataset. Given a 3D point $\mathbf{x}$ in Velodyne coordinates, we project it to a point $\mathbf{y}$ in the i-th camera image as follows:

$$\mathbf{y} = \mathbf{P}_{\text{rect}}^{(i)} \, \mathbf{R}_{\text{rect}}^{(0)} \, \mathbf{T}_{\text{velo}}^{\text{cam}} \, \mathbf{x}, \tag{15}$$

where $i = 2$, $\mathbf{P}_{\text{rect}}^{(i)}$, $\mathbf{R}_{\text{rect}}^{(0)}$, and $\mathbf{T}_{\text{velo}}^{\text{cam}}$ are calibration parameters provided in the KITTI dataset.

Additionally, we calculate the distance deviation in terms of latitude and longitude for the 3D point with respect to the origin of the IMU/GPS coordinate system in the world coordinate system. This is represented as $\mathbf{x}_{\text{GPS}}$:

$$\mathbf{x}_{\mathbf{GPS}} = \mathbf{R}_{\text{imu}}^{\text{w}} \, \mathbf{T}_{\text{velo}}^{\text{imu}} \, \mathbf{x}. \tag{16}$$



(a) Example 1 (b) Example 2 (c) Example 3

Figure 11: Three examples of labels created for the KITTI dataset [2]. The red dots indicate the location of the same point in the front view, bird's-eye view, and satellite view, respectively.

With the above process, we compute the latitude and longitude deviations for a particular 3D point relative to the GPS sensor, and determine its corresponding GPS value using the meter-to-GPS calculation method. Finally, we utilize the homography transformation $H_{\text{final}} = H_2 H_1 H$ obtained from Section A.2 to map the pixel coordinates $\mathbf{y}$ to their corresponding pixel coordinates on the BEV image. Some examples are shown in Figure 11. During model inference, we can obtain the GPS coordinates of a particular pixel on the BEV image. Using the inverse process of the above method, we can obtain the latitude and longitude deviations of the corresponding 3D point relative to the GPS sensor, and estimate the GPS value corresponding to the camera.

C Future Research

In our projection of the panoramic images into a bird's-eye view, we place a tangent plane at the south pole of the spherical imaging plane as the new imaging plane. However, this method assumes that the ground camera is oriented vertically, which may not always be the case due to roll and pitch deviations. Assuming (α, β, γ) represent the angles of $(roll, pitch, yaw)$, the rotation matrix can be obtained as follows:

$$\mathbf{R} = \begin{bmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{17}$$

By multiplying the rotation matrix $\mathbf{R}$ with the camera coordinates obtained from Equation 9, we obtain the new coordinates (x_1', y_1', z_1') in the camera coordinate system. We then proceed with the subsequent calculations to obtain the bird's-eye view image under the assumption of the specified $(roll, pitch, yaw)$ angles. Different angle configurations yield different bird's-eye view images, as illustrated in Figure 12. It can be observed that the resulting BEV appearance varies significantly when different $(roll, pitch)$ angles are given. Therefore, we suggest that in future research, $(roll, pitch)$ can be treated as additional predicted outputs to obtain more degrees of freedom in estimating the ground camera pose. Our differentiable spherical transform provides a possibility for this idea.

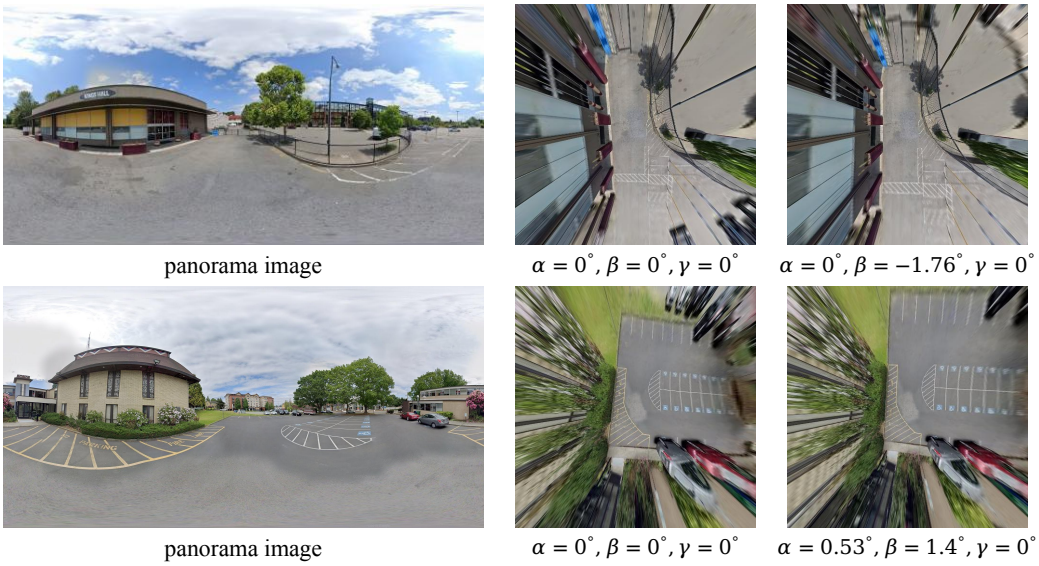


Figure 12: Bird's-eye view images obtained under different roll and pitch angles using our spherical transform method.

References

[1] https://en.wikipedia.org/wiki/Web_Mercator_projection#References.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[3] Ted Lentsch, Zimin Xia, Holger Caesar, and Julian FP Kooij. Slicematch: Geometry-guided aggregation for cross-view pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[4] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3640–3649, June 2021.