2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Renderable Street View Map-Based Localization: Leveraging 3D Gaussian Splatting for Street-Level Positioning

Howoong Jun[1], Hyeonwoo Yu[2], and Songhwai Oh[3]

***Abstract*— In this paper, we introduce a new method that first utilizes 3D Gaussian splatting in street-level localization problem. Robust localization with street-level real-world images such as street view is a major issue for autonomous vehicle, augmented reality (AR) navigation, and outdoor mobile robots. The objective is to determine the position and orientation of a query image that matches a street view database composed of RGB images. However, given the limited information available in the street view images, accurately determining the location solely based on this data presents a significant challenge. To address this challenge, we propose a novel method called renderable street view map-based localization (RSM-Loc). This approach enhances the localization process by augmenting 2D street view images into a renderable 3D map using 3D Gaussian splatting, to resolve street-level localization problems. Upon receiving a query RGB image without geometry information, the proposed method renders 2D images from a pre-made renderable map and compares image pose similarities between the rendered images and the query image. Through iterations of this process, the proposed method eventually estimates the pose of the given query image. The experimental results demonstrate that RSM-Loc outperforms the baselines with neural-field-based localization. Additionally, we conduct deep analysis on the proposed method to show that our method can serve as a new concept for the street-level localization problem.**

I. Introduction

Street-level localization is one of the key problems across many applications such as autonomous vehicle [1] and augmented reality (AR) navigation for mobile phones [2]. While GPS sensors are commonly employed for outdoor positioning, their accuracy diminishes in areas with 'urban canyons' like Manhattan, New York, where tall buildings can block or distort signals [3], making precise location determination difficult. Mostly, many methods utilize high definition (HD) maps which include detailed 3D information of urban scenes such as lanes, traffic lights, and traffic signs to overcome this issue. However, creating an HD map requires high-quality processors and sensors such as 3D LiDAR, which are expensive both in terms of cost and computational requirements [4]. Therefore, a new approach that can function even with low-precision data is required.

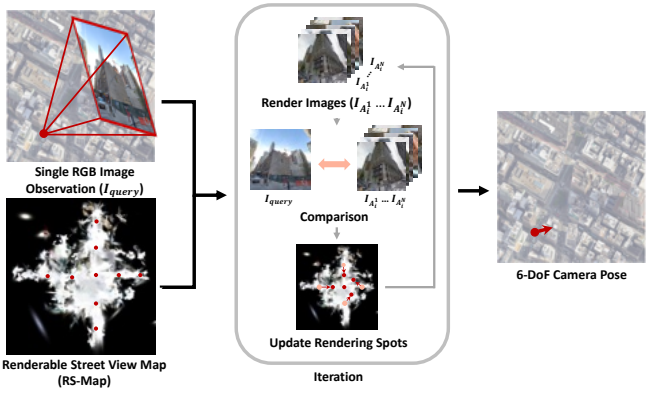

Fig. 1. An overview of the proposed method. The objective of the method is to determine the 6-DoF pose of an input query image based on the renderable street view map.

Recently, some researches focus on utilizing street view images with standard definition (SD) maps without any 3D information of the scene [5]–[8]. Mostly, these methods focus on finding correspondence between street view images and pre-collected SD map patches. Even though these studies require low computational burden and memory compared to the HD map-based methods, they are unable to estimate precise location since SD map is expressed in an abstracted format. Also, they are vulnerable in dynamic objects such as cars and pedestrians which are not considered in the map.

To overcome these issues, we propose a new concept of street-level localization method by using 3D Gaussian splatting [9] named renderable street view map-based localization (RSM-Loc). RSM-Loc is the first street-level localization method that utilizes 3D Gaussian splatting scene representation to street view images. RSM-Loc aims to estimate the position and orientation of a RGB image using a renderable street view map (RS-Map) generated by 3D Gaussian splatting. Our approach does not necessitate any pre-existing 3D geometric data, such as HD maps; instead, it solely relies on street view image data. To avoid noisy pixels of reconstructed image, we use key point extraction and matching to compare significant pixels. With the matching results, the proposed method actively search to estimate its position. Figure 1 shows the overview of RSM-Loc.

We demonstrate RSM-Loc in StreetLearn dataset [10] and

*This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning).

[1]Howoong Jun is with Interdisciplinary Program in Artificial Intelligence (IPAI), Seoul National University and Automation and Systems Research Institute (ASRI), and Sequor Robotics Inc., Seoul, Korea (Republic of) howoong.jun@rllab.snu.ac.kr

[2]Hyeonwoo Yu is with Department of Intelligent Robotics & Mechanical Engineering, Sungkyunkwan University (SKKU), Suwon-si, Gyeonggi-do, Korea (Republic of). hwyu@skku.edu

[3]Songhwai Oh is with the Department of Electrical and Computer Engineering (ECE) & Interdisciplinary Program in Artificial Intelligence (IPAI), Seoul National University and Automation and Systems Research Institute (ASRI), and Sequor Robotics Inc., Seoul, Korea (Republic of) songhwai@snu.ac.kr

Corresponding author: Songhwai Oh.

compare it with existing 3D rendering-based localization methods. To evaluate our approach, we use two datasets that covers the streets of Manhattan, New York: StreetLearn Manhattan and Touchdown [11]. We divide the datasets into two categories for map construction and queries. The Manhattan dataset is utilized for constructing the RS-Map, while the Touchdown dataset is employed to supply query images for validating localization problem.

Overall, the contributions of our work include:

- We propose RSM-Loc, the first work that uses 3D scene representation for street-level real-world image-based localization task.
- The proposed method resolves the problem of monocular visual localization only with RGB images without using any key frames that have both 2D and 3D features.
- The paper presents the concept of 'render, compare, and update' to address the challenge of image-based localization.

II. Related Work

1) Street-Level Localization: Street-level localization with street view image methods have been frequently studied recently. [5] proposes a set of 16 dimensional vectors named embedded space (ES) feature which represents corresponding image and map tile pairs. With the power of deep learning, the method estimates the most similar top-N map tiles by four street view images given. [6] utilizes the feature of [5] for street-level tracking problem. They apply ES features of multiple map tiles as observations in Monte Carlo localization to prove its efficiency. OrienterNet [7] proposes neural map matching which matches gravity-aligned input image and GPS-priored SD map. They transform both input image and SD map into the same domain and conduct BEV-map matching. SNAP [8] improve the concept of neural map proposed in OrienterNet into multi-modal neural map with multiple street view images and satellite images. Due to this improved neural map, the method can cover broader region than OrineterNet.

2) 3D Rendering on Street View: Researchers have been developing methods for 3D reconstruction with street view images. Block-NeRF [12] suggests reconstructing block-by-block individually with NeRF [13]. This method enables NeRF to be expanded in broad regions such as streets. Matrixcity [14] make synthetic city dataset named matrixcity for city scale NeRF research. Simulation-based environment makes it possible to get dynamic scenes from ground to aerial, and their ground-truth camera poses. With the recent advancement in 3D Gaussian splatting [9], many researchers are trying to apply this technology to reconstruct realistic street scenes. DrivingGaussian [15] introduces a method for modeling complex driving scenes with two modules: incremental static 3D Gaussians and composite dynamic Gaussian graphs. These two modules enable the description of both static and dynamic urban driving scenes with no blur. Street Gaussian [16] divide scene representation into point-based background model and foreground object model. This approach make it possible to decompose foreground objects for editing applications.

3) Neural Field-Based Localization: With the rise of 3D rendering technology such as neural radiance fields (NeRF), researchers try to apply them to solve localization problem. Loc-NeRF [17] apply NeRF [13] and Monte Carlo localization method to estimate the position of moving robots. They compare pixel-by-pixel between reconstructed image created with NeRF and query image and use the pixel similarity to solve the problem. Since simply comparing pixels in images does not guarantee performance due to the noise present in the reconstructed images, they apply tracking methods to address the issue. However, there still exists ambiguous result of the method since comparing query image with noisy reconstructed image can derive high computational cost result which is undesirable for the problem. Also, this approach requires multiple sequential images for accurate localization since the method utilizes tracking.

III. Methods

The main purpose of RSM-Loc is to find the position of the street-level RGB input image I_{query} using a renderable map created by street view images. For this purpose, we utilize 3D Gaussian splatting [9] for composing the RS-Map. Utilizing the RS-Map, the proposed method renders images on specific locations on the RS-Map referred to as anchors, and extracts significant pixels on the rendered images. By iteratively comparing these significant pixels between the rendered images and I_{query}, and updating the anchors, RSM-Loc ultimately identifies the scene within the map that corresponds to I_{query}. The overall process for RSM-Loc is described in Figure 2. We assume that the approximate location within a radius of 100 meters is known beforehand, through GPS or other means.

A. 3D Renderable Map

The data that are used for composing the RS-Map are RGB image and its latitude, longitude, and orientation. We define a set of latitude, longitude, and orientation in this data as initial anchors $\{A_1^1, ..., A_1^N\} \in \mathbb{A}$ and compose 3D renderable map as RS-Map $\mathbb{M}_{RS} \in \{\mathbb{R}^{H_{map} \times W_{map} \times L_{map}}, \mathbb{A}\}$. H_{map}, W_{map}, and L_{map} stand for the height, width, and length of 3D Gaussisan splatting training result, respectively. To sum, RS-Map consists of two types of data: renderable data based on 3D Gaussian splatting and initial anchors.

Given that the proposed method operates under the assumption of a prior approximation within a maximum radius of 100 meters, we divide the dataset into blocks, following the approach outlined in [12], to construct the RS-Map. Centering on each intersection, we define initial anchors on the roads extending in four directions. In this context, the RS-Map covers an area with a maximum radius of 100 meters. The StreetLearn dataset consists of equirectangular images, which we transform into eight directional images with a 45-degree angle. Furthermore, to capture the skyline of the city without interference from on-road obstacles such

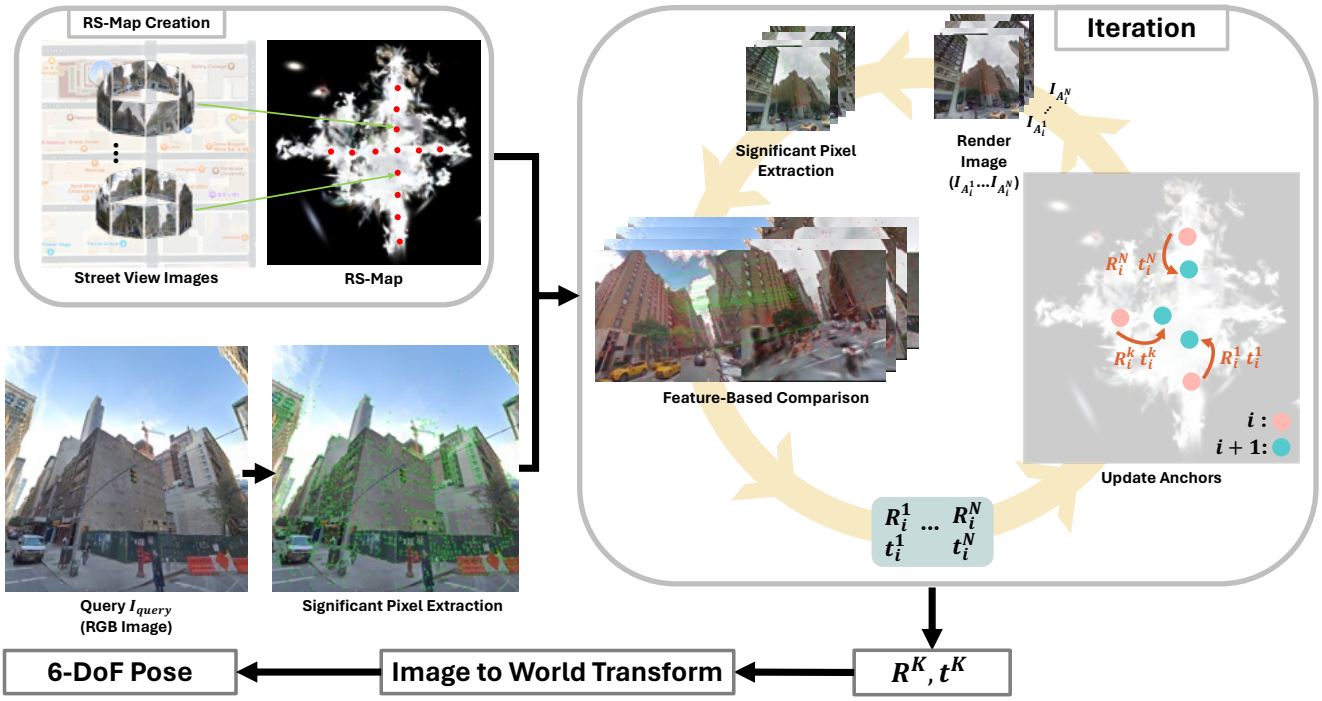

Fig. 2. An overall process of the proposed method. To construct a renderable street view map (RS-Map), the proposed method uses 3D Gaussian splatting and searches for the pose that matches the given query image. We start by placing anchors at initial positions. Poses of each anchor are updated by comparing the query image with the images rendered from their pose. Since the rendered images include noisy pixels, we select significant pixels for the comparison by using local feature extraction.

as pedestrians and cars, we utilize upper-view images angled at 30 degrees for map building.

B. Localization

The localization process consists of three steps: rendering, comparison, and updating. In other words, our approach entails rendering images at the anchor points, followed by comparing visual features between these rendered images and the query image. Subsequently, RSM-Loc updates the 3D pose of the anchors sequentially.

In the rendering step, the anchor poses are initialized on RS-Map $A_1^1, ..., A_1^N$ which are distributed on the road. This ensures that RSM-Loc does not commence from inappropriate initial positions, such as within a building. RSM-Loc starts by rendering images at the initial positions and compare the rendered images $I_{A_1^1}^1, ..., I_{A_1^N}^1$ with the input query image I_{query}.

In the comparison step, there exists some noises on the rendered images due to the quality of SfM for training 3D Gaussian splatting. In this regard, evaluating each pixel individually entails considering unwanted pairs, including those resulting from noise. To circumvent this issue, we extract significant pixels by exploiting visual key point detection, and perform key point matching to compare two images. We use Superpoint [18] with Brute-force algorithm for comparing two images. Using robust significant pixels allow for the exclusion of noisy pixels when comparing the query image with the rendered images, focusing solely on meaningful pixels that contain information for the comparison step. Despite inaccuracies, noises, or missing data in the 3D renderable map scenes, our method can localize itself without relying on a perfect 3D reconstruction.

In the update step, the pose of the anchors are adjusted based on the estimated pose $[\mathbf{R}|\mathbf{t}]$ with the matched features.

$$\mathbf{t} = \mathbf{C_t} \odot \mathbf{UW\Sigma U}^T, \tag{1}$$

$$\mathbf{R} = \mathbf{UW}^{-1}\mathbf{V}^T, \tag{2}$$

where $\mathbf{C_t}$ is a scaling factor for translation, and $\odot$ denotes element-wise multiplication. $\mathbf{U}$ and $\mathbf{W}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix derived by singular value decomposition during pose estimation. In this way, we can determine the values of rotation $R_i^1, ..., R_i^N$ and translation $t_i^1, ..., t_i^N$ for updating the pose at i^{th} step where $i \in [1, M)$. A new set of anchors $A_{i+1}^1, ..., A_{i+1}^N$ is created after the update step. By repeatedly rendering images, comparing, and updating for M steps, the anchors eventually converges to the scene of the I_{query}. In the end, we can obtain the final estimated transformation $[R^k|t^k]$ from initial pose to final pose of k^{th} anchor as follows:

$$R^k = R_1^k \cdot R_2^k \cdot ... \cdot R_{M-2}^k \cdot R_{M-1}^k, \tag{3}$$

$$t^k = R_1^k t_1^k + R_2^k t_2^k + ... + R_{M-2}^k t_{M-2}^k + R_{M-1}^k t_{M-1}^k, \tag{4}$$

where $k = 1, ..., N$ which are anchor indices. To derive the final pose estimation result, we select the anchor A_M^K within the least relative error in rotation compared to I_{query} as the ultimate outcome. Note that only the relative rotation and translation between the two images are the metrics used

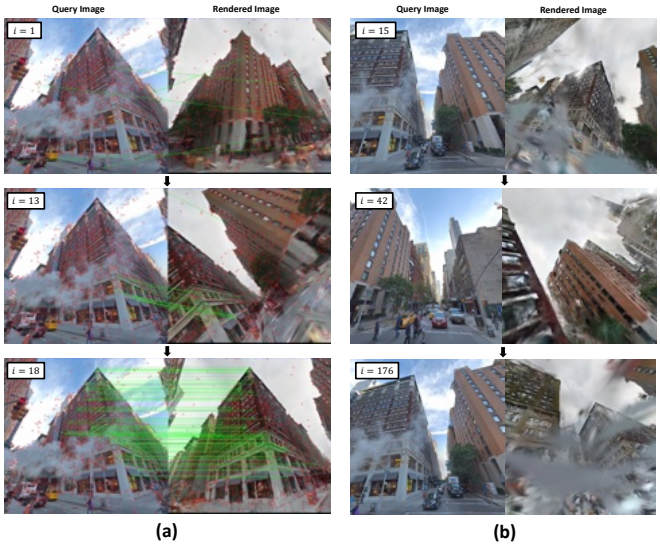

Fig. 3. Qualitative results of RSM-Loc and Loc-NeRF. Left columns of (a) and (b) are query images from Touchdown dataset and right columns are rendered images. (a) Qualitative result of RSM-Loc (b) Qualitative result of Loc-NeRF with particle filter.

to determine the validity of the outcome. Additionally, to expedite the process, we terminate it before reaching step M if the relative rotation and translation meet the threshold.

However, since the derived transformation $[R^K|t^K]$ is based on the image coordinate, we have to translate the transformation into the world coordinate. As the initial anchors of the RS-Map contain the initial latitude, longitude, and orientation on the world coordinate, we can derive the pose matrices on the world coordinate $([{}^{\mathbf{w}}R_{A_1^1}|{}^{\mathbf{w}}t_{A_1^1}], ..., [{}^{\mathbf{w}}R_{A_1^N}|{}^{\mathbf{w}}t_{A_1^N}])$ using the RS-Map. Using the derived transformations from equation 3 and 4 on the image coordinate, we can use the following equation to compute the final world-coordinate pose from the selected anchor A_1^K:

$$R_{final} = {}^{\mathbf{w}}_{\mathbf{I}}R \cdot {}^{\mathbf{I}}_{\mathbf{I}}R^K \cdot {}^{\mathbf{I}}_{\mathbf{w}}R \cdot {}^{\mathbf{w}}R_{A_1^K} \tag{5}$$

$$t_{final} = t_{A_1^K} + \alpha \cdot {}^{\mathbf{w}}_{\mathbf{I}}R \cdot {}^{\mathbf{I}}t^K, \tag{6}$$

where a transform matrix from world to image is

$${}^{\mathbf{w}}_{\mathbf{I}}R = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}, \tag{7}$$

and ${}^{\mathbf{I}}_{\mathbf{w}}R$ is an inverse of ${}^{\mathbf{w}}_{\mathbf{I}}R$. The scale parameter α represents the ratio of distances between the world coordinate and the image coordinate, which can be calculated by

$$\alpha = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{||{}^{\mathbf{w}}t_{A_1^i} - {}^{\mathbf{w}}t_{A_1^j}||_2}{||{}^{\mathbf{I}}t_{A_1^i} - {}^{\mathbf{I}}t_{A_1^j}||_2}, \forall i \neq j. \tag{8}$$

$||{}^{\mathbf{w}}t_{A_1^i} - {}^{\mathbf{w}}t_{A_1^j}||_2$ and $||{}^{\mathbf{I}}t_{A_1^i} - {}^{\mathbf{I}}t_{A_1^j}||_2$ represent Euclidean distances between two initial anchor points on the world coordinate and on the image coordinate, respectively.

In summary, the final pose on the world coordinate is obtained by transforming the pose of the anchor with the smallest relative rotation error on the image coordinate, as derived from Equation 3, to the world coordinate using Equation 5 and 6.

IV. Experiments

A. Dataset

We apply RSM-Loc into Manhattan and Touchdown dataset from StreetLearn. We opt for Manhattan, New York as our testing location due to its abundance of tall buildings, facilitating an evaluation of pose estimation encompassing not only position and heading but also roll and pitch orientation. Also, an 'urban canyon' environment like this serves to validate the necessity of the proposed method. Both datasets cover Manhattan in New York, but the images were taken at different times, location, and under different circumstances such as weather and traffic. We select intersections that are surrounded by tall buildings, which is characteristic of urban canyons, for our experiments. To address the visual localization problem using our proposed method, we utilize the Touchdown dataset as a source of query images (validation set) and the Manhattan dataset for constructing the RS-Map.

The datasets contain 1664×832 sized equirectangular panorama RGB images, position (latitude and longitude), altitude, orientation (roll degree, pitch degree, and heading degree) of the panoramic camera, list of directly connected neighbors, and captured date. We utilize equirectangular panorama RGB image, position, and orientation for creating RS-Map. We split the equirectangular panorama image into eight RGB images with 45-degree angle, each with a size of 832×832. In order to leverage permanent features such as buildings in street view, we utilized images with a 30-degree roll on the image coordinates. Also, since both Manhattan and Touchdown datasets in StreetLearn aim to solve language ground tasks for agents in street environments [11] they do not provide camera transformation matrix for the datasets. Therefore, we estimate the transformation from world to image as shown in Equation 7.

B. Qualitative Results

Figure 3a illustrates an example of RSM-Loc episode. The query image on the left side is taken from the Touchdown dataset, and the rendered image on the right side is generated from RS-Map. The episode begins with a query image and a rendered image that do not share any common features ($i = 1$). RSM-Loc searches for rendered scene that overlaps with the query image. Following movements based on the matching results, the proposed method identifies overlapping scenes ($i = 13$) and aligns itself within the scene ($i = 18$). Our method avoids the problem of blur noises in the rendered scene by extracting the significant pixels. The blocking noises caused by fumes in the query image, which the rendered image does not have, do not affect our method either. Eventually, the proposed method successfully locates its pose that matches the query image. To compare with Loc-NeRF, as shown in Figure 3b, we use a radius of 10 meters,

TABLE I

TRANSLATION AND ROTATION ERROR

Error	Radius	**RSM-Loc (Proposed)**				Loc-NeRF [17] (Single Image)				Loc-NeRF [17] (Particle Filter)			
		100 m	50 m	20 m	10 m	100 m	50 m	20 m	10 m	100 m	50 m	20 m	10 m
Translation (m)	Mean ↓	**23.70**	**15.15**	8.27	7.21	56.42	30.19	13.27	6.90	23.77	18.67	**7.54**	**2.55**
	Median ↓	**11.32**	**9.73**	**7.29**	6.57	55.55	30.50	14.10	6.95	24.17	16.20	7.35	**2.42**
Rotation (rad)	Mean ↓	**0.68**	**0.67**	**0.30**	**0.37**	1.68	1.63	1.75	1.82	2.40	2.35	2.18	2.24
	Median ↓	**0.10**	**0.09**	**0.07**	**0.06**	1.62	1.59	1.67	1.69	2.60	2.36	2.29	2.22
Heading (rad)	Mean ↓	**0.52**	**0.45**	**0.22**	**0.26**	2.85	2.75	2.76	2.68	2.54	2.52	2.36	2.26
	Median ↓	**0.04**	**0.04**	**0.02**	**0.03**	2.38	2.34	2.33	2.34	2.60	2.48	2.26	2.19

as Loc-NeRF is designed for areas under 10 meters. Since particle filter is based on the sequential multiple images, query images change in the result as described in the left column.

C. Quantitative Results

We follow the standard evaluation metric for position and orientation accuracy as [19]. For the positional error, we measure Euclidean distance of estimated position p_{est} and ground truth position p_{gt} as $||p_{est} - p_{gt}||_2$. To measure the orientation error, we compute minimum angle θ that satisfies the equation below:

$$2cos(|\theta|) = tr(\mathbf{R}_{gt}^{-1}\mathbf{R}_{est}) - 1, \tag{9}$$

where $\mathbf{R}_{gt}$ is ground truth rotation matrix calculated from orientation data of the dataset and $\mathbf{R}_{est}$ is estimated rotation matrix derived by RSM-Loc. We also examine the heading-only error, which is a crucial factor for street-level localization problem. The overall quantitative results of the experiments are shown in Table I.

We compare our method with neural-field-based localization method, Loc-NeRF [17]. Loc-NeRF employs a particle filter, necessitating multiple sequential images, whereas RSM-Loc is a single-shot localization method. We first compare pose error with likelihood from the 'measurement step' used in Loc-NeRF. In this case, we select one image with the lowest weight calculated using the method. Additionally, we compare Loc-NeRF integrated with particle filter. For the prediction step of the particle filter, ground truth odometry is provided to Loc-NeRF. In Table I, Loc-NeRF demonstrates translation errors that are either less than or comparable to those of RSM-Loc in small areas under 20m. However, it fails to correctly estimate its orientation, which is indeed a wrong estimation. We vary the radius of the testing area from 10m to 100m. Note that the median rotation error for RSM-Loc remains below 6 degrees across all testing areas, while the median heading error remains under 3 degrees, regardless of the testing area. As Loc-NeRF is only designed for small indoor areas of 1 $\times$ 0.5 $\times$ 3.5m, its efficacy decreases with larger testing areas.

D. Ablation Study

1) Initial State with or without Overlapping Scenes: We perform extra experiments to demonstrate that RSM-Loc can locate itself irrespective of the overlap portion. To test the performance of RSM-Loc, we change the heading value of the initial state from 0 to 180 degrees and see if it can match the given query image with the correct scene. Figure 4 shows the result of the ablation study. The correct scene is found for both overlapping scenes, such as those with an initial heading of 0 or 45 degrees, and non-overlapping or less-overlapping scenes, such as those with an initial heading of 90, 135, or 180 degrees. Since our method is based on the 3D Gaussian splatting, there exists some 3D Gaussian-shaped noises on the rendering images. Despite the presence of noises on the rendered images, the significant pixel extraction process effectively avoids utilizing these noises and accurately identifies the pose. Furthermore, in overlapping scenes (0 and 45 degrees), the proposed method fine-tunes translation to precisely match into the query image.

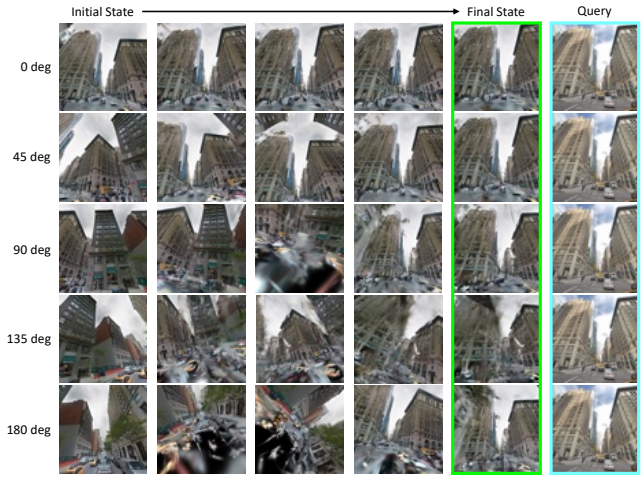

Fig. 4. Ablation study for RSM-Loc. Given a query image, we vary the heading value of the initial state from 0 to 180 degrees and test whether RSM-Loc is able to find its final pose that corresponds to the query image. The proposed method can successfully locate the scene of the query image, regardless of what its initial state is.

2) Robustness on Environmental Change or Rendering Noise: Additionally, we analyze the results of our method to demonstrate the effectiveness of the proposed approach. Figure 5 serves to visually represent the qualitative analysis of RSM-Loc. Figure 5a shows the results of environmental changes between the dataset used for query image (Touchdown) and the dataset employed for constructing the RS-Map (Manhattan). In the first two rows, the query image has buildings on the red box whereas rendered image from RS-Map does not have. The third row illustrates the weather

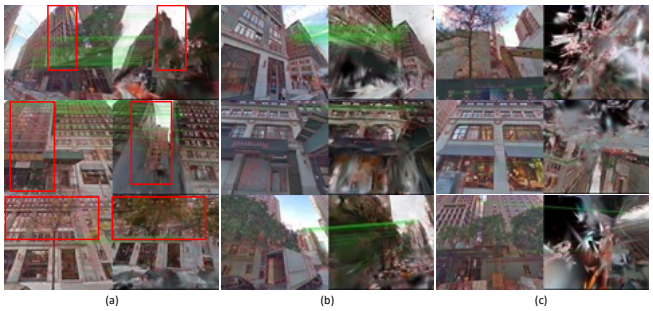

Fig. 5. Additional results of RSM-Loc. (a) Environmental changes between query image and mapping image. The first and second rows indicate the presence of a new building, while the third row indicates the presence of leaves depending on the seasonal variation. (b) Localization on noisy rendering results. Even in the presence of noise in the rendered images, the proposed method yields correct results by comparing only the valid parts of the images. (c) Failure case of RSM-Loc. During the update step, if rendering occurs from too disparate viewpoints, resulting in the absence of valid pixels, the process fails.

variations between the scenes in the Touchdown dataset and the Manhattan dataset. Despite these environmental changes, RSM-Loc successfully aligns with the query image. The rotation errors for first, second, and third row are 3.81, 3.67, and 9.87 degrees, respectively. The translation errors are 6.87, 10.64, and 5.07 meters, and heading errors are 0.81, 2.11, and 10.43 degrees, respectively. Figure 5b demonstrates the impact of rendering quality on RSM-Loc. Despite lower rendering quality, the proposed method accurately determines its pose by considering only the valid parts of the image. The rotation errors for first, second, and third row are 2.77, 1.15, and 9.92 degrees, and translation errors are 7.54, 6.87, and 9.04 meters, respectively. The heading errors are 0.59, 0.01, and 0.23 degrees, respectively. This demonstrates that the effectiveness of the proposed method does not rely on a perfect rendering process.

3) Failure Cases: Figure 5c illustrates a failure case of RSM-Loc. If the viewpoint observes invalid areas such as the ground or inverted scenes, as depicted in the right column of the Figure 5c, a scene mixed with black and 3D Gaussian noise is displayed. In this case, since there are no valid parts, the proposed method fails in estimating the position.

V. Conclusion

In this paper, a new street-level localization method with 3D Gaussian splatting named RSM-Loc is proposed. RSM-Loc is the first work that utilizes the advantages of 3D Gaussian splatting methods into street-level localization problem. To filter out noisy pixels of the reconstructed images, we utilize key point detection to extract significant pixels and key point matching for comparing the images. The experiments demonstrate that the proposed method achieves high performance in solving the localization problem with real-world images compared to the neural-field-based localization method. Additionally, we deeply analyze our method on diverse situations to show that our method can be a new concept for street-level real-world visual localization.

References

[1] D. Liu, Y. Cui, X. Guo, W. Ding, B. Yang, and Y. Chen, "Visual localization for autonomous driving: Mapping the accurate location in the city maze," in *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2020.

[2] S. Yan, Y. Liu, L. Wang, Z. Shen, Z. Peng, H. Liu, M. Zhang, G. Zhang, and X. Zhou, "Long-term visual localization with mobile sensors," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 245–17 255.

[3] H. Lee, J. Seo, and Z. Z. M. Kassas, "Urban road safety prediction: A satellite navigation perspective," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 94–106, 2022.

[4] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[5] N. Samano, M. Zhou, and A. Calway, "You are here: Geolocation by embedding maps and images," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.

[6] M. Zhou, X. Chen, N. Samano, C. Stachniss, and A. Calway, "Efficient localisation using images and openstreetmaps," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[7] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulò, R. Newcombe, P. Kontschieder, and V. Balntas, "Orienternet: Visual localization in 2d public maps with neural matching," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] P.-E. Sarlin, E. Trulls, M. Pollefeys, J. Hosang, and S. Lynen, "Snap: Self-supervised neural maps for visual positioning and semantic understanding," in *Proc. of the Neural Information Processing Systems (NeurIPS)*, 2023.

[9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[10] P. Mirowski, A. Banki-Horvath, K. Anderson, D. Teplyashin, K. M. Hermann, M. Malinowski, M. K. Grimes, K. Simonyan, K. Kavukcuoglu, A. Zisserman, *et al.*, "The streetlearn environment and dataset," *arXiv preprint arXiv:1903.01292*, 2019.

[11] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proc. of the Eurpoean Conference on Computer Vision (ECCV)*, 2020.

[14] Y. Li, L. Jiang, L. Xu, Y. Xiangli, Z. Wang, D. Lin, and B. Dai, "Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[15] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[16] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.

[17] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[18] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[19] A. Jafarzadeh, M. L. Antequera, P. Gargallo, Y. Kuang, C. Toft, F. Kahl, and T. Sattler, "Crowddriven: A new challenging dataset for outdoor visual localization," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.