OmniColor: A Global Camera Pose Optimization Approach of LiDAR-360Camera Fusion for Colorizing Point Clouds

Bonan Liu[1*], Guoyang Zhao[1*], Jianhao Jiao[1], Guang Cai[2], Chengyang Li[1],
Handi Yin[1], Yuyang Wang[1], Ming Liu[1,2] and Pan Hui[1,2]

Abstract— A Colored point cloud, as a simple and efficient 3D representation, has many advantages in various fields, including robotic navigation and scene reconstruction. This representation is now commonly used in 3D reconstruction tasks relying on cameras and LiDARs. However, fusing data from these two types of sensors is poorly performed in many existing frameworks, leading to unsatisfactory mapping results, mainly due to inaccurate camera poses. This paper presents Omni-Color, a novel and efficient algorithm to colorize point clouds using an independent 360-degree camera. Given a LiDAR-based point cloud and a sequence of panorama images with initial coarse camera poses, our objective is to jointly optimize the poses of all frames for mapping images onto geometric reconstructions. Our pipeline works in an off-the-shelf manner that does not require any feature extraction or matching process. Instead, we find optimal poses by directly maximizing the photometric consistency of LiDAR maps. In experiments, we show that our method can overcome the severe visual distortion of omnidirectional images and greatly benefit from the wide field of view (FOV) of 360-degree cameras to reconstruct various scenarios with accuracy and stability. The code will be released at https://github.com/liubonan123/OmniColor/.

I. INTRODUCTION

Over the last two decades, various reality capture methods have been successfully designed for reconstructing large-scale environments by using cameras, LiDAR, as well as other perceptual sensors. Camera-based photogrammetry methods [1], [2] typically extract visual features from textured scenes and then utilize the overlapping areas to align photos taken from different angles. However, these methods are highly susceptible to variations in illumination and visual complexity of scenes. LiDAR-based scanning methods [3], [4], which typically extract structural features from point clouds, are invariant to such changes. However, a fundamental drawback of LiDAR is that they do not provide rich visual appearance information, which is exceptionally beneficial for humans to recognize geometric information from enormous amounts of point clouds. Recently, some systems [5], [6], [7], [8], [9] have been designed to utilize the complementary properties of LiDARs and cameras to create colored point cloud maps but still suffer from blurring,

*Indicates Equal Contribution.

[1]B. Liu, G. Zhao, J. Jiao, C. Li, H. Yin, Y. Wang, IEEE Member, M. Liu, IEEE Senior Member, P. Hui, IEEE Fellow, are with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

[2]G. Cai is with The Hong Kong University of Science and Technology, Hong Kong SAR, China

E-mail: {bliu404, gzhao492, cli386, hyin335}@connect.hkust-gz.edu.cn, jiaojh1994@gmail.com, gcaiab@connect.ust.hk, {yuyangwang, eelium, panhui}@ust.hk

Fig. 1. The RGB-Colored point cloud reconstructed by our mobile mapping system. A fly-through rendered image from the point cloud's left and right sides.

ghosting, and other visual artifacts, due to inaccurate camera poses and illumination variations.

To mitigate the problems above, an effective strategy is to enlarge the camera field of view (FOV) [10]. The expansion of FOV facilitates capturing additional scene information, broadening the co-visible regions among successive frames. It enables the system to capture the surrounding area simultaneously to reduce the artifacts from illumination variation and acquire sufficient correspondence for improved adaptive ability in diverse environments. Concurrently, progress in lens manufacturing techniques has made it possible to attain complete 360-degree perception with only two lenses mounted back-to-back [11], providing a cost-effective mobile mapping solution.

For customized handheld mobile mapping devices with an extra commercial 360-degree camera [12], several issues may negatively affect the quality of reconstructed colored point clouds. First, the input geometric model [13] for which the colored point cloud is constructed is inaccurate and produced from noisy data. Second, LiDARs and cameras are not hardware-synchronized. The color images are not in rigid correspondence with the LiDAR range measurements. Third, panorama images are subject to significant visual distortion caused by the spherical projection process. Lastly, establishing correct correspondences among data in heterogeneous modalities (*e.g.,* geometry and texture) is challenging. For instance, several systems [14], [15] that rely on 2D-3D geometric line correspondences often encounter misalignment issues between images and LiDAR-reconstructed maps.

In this paper, we describe an approach for optimizing the mapping of color images produced by an independent 360-

degree camera to a corresponding geometric reconstruction. Our method addresses the above problems through a cohesive optimization framework, wherein we jointly optimize the poses of all color frames to rectify inaccurate colorization regions. The camera poses for all images are globally optimized to maximize a unified objective: the photometric consistency of the colored point cloud maps. Furthermore, our approach is specially adjusted and highly suitable for 360-degree cameras. It allows us to bypass the non-differentiable changes in the visibility relationship between the camera and point cloud during optimization, significantly reducing the computational cost. Overall, our contributions can be summarized as follows:

- We propose a novel global optimization approach of LiDAR-360 camera fusion for convenient and precise point cloud colorization (see Fig. 1), which can overcome the severe visual distortion in omnidirectional images and gain benefit from a wider FOV.
- We propose a novel point cloud co-visibility estimation approach, which mitigates the impact of the noise in point cloud surfaces on visibility relationships.
- Our approach operates in a readily available manner, enabling seamless integration with any mobile mapping system while ensuring both convenience and accuracy. Extensive experiments demonstrate its superiority over existing frameworks.

II. RELATED WORKS

A. Feature-based LiDAR-Camera Alignment Methods

With the emergence of non-repetitive LiDAR, it is possible to obtain increasingly denser point clouds over time at stationary localizations, which can acquire more precise and distinct edge features from the environment, thereby facilitating the development of targetless LiDAR-camera calibration methods. In certain studies, [14], [16] have proposed leveraging inherent 3D and 2D edges extracted from LiDAR maps and camera images, and [15] designs an edge alignment method based on the edge feature of LiDAR intensity images. Extrinsic parameters are optimized by minimizing the distance between corresponding edge features in both sensors. Subsequently, each point cloud is transformed into the camera frame and projected onto the image plane to obtain corresponding RGB information. However, the colored point clouds are the indirect outcome of these methods. All these methods are designed for LiDAR-camera extrinsic calibration, which works only in edges' richness in natural scenes.

In addition, CP+ [12] introduces a series of preprocessing operations, including removing blurry images and dynamic objects and selecting an appropriate region of interest (ROI) from the point cloud to address misalignment issues. However, feature mismatches and the absence of global constraints make these feature-based LiDAR-camera alignment methods unreliable in unstructured and textureless environments [6].

B. Motion-based LiDAR-Camera State Estimation Solutions

Paintcloud [5] firstly proposes an offline method for point cloud colorization in a plug-and-play manner, and it can be seamlessly integrated with any LiDAR Odometry (LO) system [3], [4]. However, the camera poses used in this method are calculated by linear interpolation with LiDAR trajectory, which leads to poor colorization results. By simply combining with the recent LiDAR-Visual-Inertial Odometry (LVIO) systems, the colorization result can be improved. LVI-SAM [17] employs a factor graph as a tightly-coupled smoothing and mapping framework for sensor fusion. R2LIVE [18] and R3LIVE [6] effectively integrate data from LiDAR-Inertial-Visual sensors by minimizing the feature re-projection error and directly optimizing the frame-to-map photometric error, respectively. In addition, V-LOAM [19] and DV-LOAM [20] fuse LiDAR-inertial-visual sensors at a level of loosely-coupled, in which the state is not jointly optimized with all sensor measurements. The aforementioned Fusion-SLAM systems are specifically designed for localization and mapping applications, with some of them having already demonstrated available dense colored point clouds [6], [21]. However, the cost function of these systems accounts for frame-to-map photometric residual without any global mapping consistency constraint consideration. Moreover, these SLAM-based online mapping techniques are unsuitable for accommodating customized high-resolution cameras in order to achieve enhanced colorization outcomes.

We draw inspiration from the photometric consistency mapping approach for mesh colorization [22]. However, this method is primarily designed for mesh representative reconstructions, limiting its applicability and robustness when extended to point clouds due to noise on the point cloud surface. We improve upon this by proposing a point cloud co-visibility estimation approach. Our algorithm is highly suitable for 360-degree cameras as it allows us to bypass the non-differentiable changes in the visibility relationship between the camera and point cloud during optimization. By seamlessly integrating high-resolution independent devices (360-degree cameras) into a mobile mapping system, our method ensures convenience and accuracy.

III. METHODOLOGY

A. Overview

Fig. 2 presents an overview of our proposed methodology, in which the input is derived from two robustly established systems: the LiDAR-Inertial Odometry (LIO) system and the Visual Odometry (VO) system. By employing LIO-based approaches [21], [13], [23], we can acquire highly accurate LiDAR poses and enough corrected point cloud maps, facilitating the generation of a smooth and precise motion trajectory using high-frequency IMU data. Additionally, [24], [25], [26] can be optionally employed as preprocessing steps to enhance the quality of the point cloud results. We denote $^L\mathbf{P}$ as the final global point cloud input. Meanwhile, we also implement a VO system [10], [27] to estimate the time offset
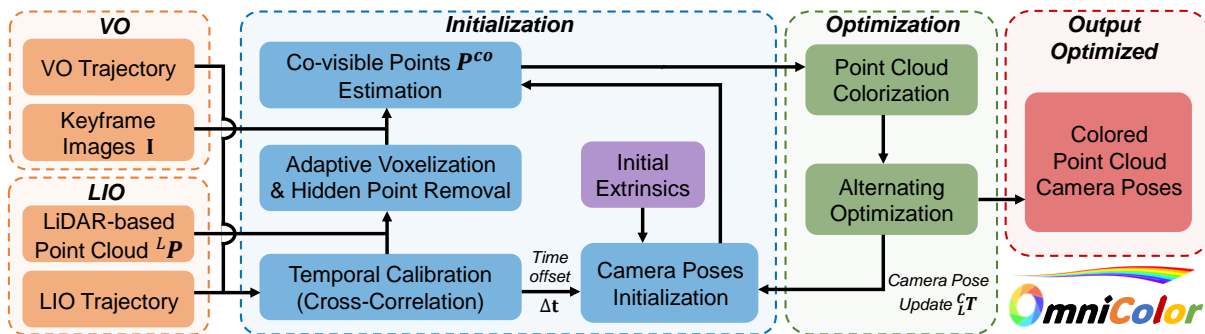
Fig. 2. Overview of our proposed system.

Δt between LiDAR and camera systems by comparing their motion signals through cross-correlation.

Given the handheld nature of the mobile mapping device, motion blur noticeably affects the quality of the input color images. In order to enhance the computed color point clouds, our pipeline autonomously chooses a subset of input images, utilizing a series of keyframes generated by the VO system as a basis for selection. Then, we evaluate and quantify the blurriness of each keyframe image using a no-reference metric [28]. Subsequently, we add the frame with the lowest blurriness within each time segment (t_-, t_+) after selecting the last keyframes. The selected image set is denoted as $\mathcal{I} = \{I_1, ...I_n\}$. The coarse keyframe poses are calculated using linear interpolation based on the initial extrinsic parameters, temporal calibration result, and device trajectory. Here, we denote n as the number of keyframes, $_L^C\mathcal{T} = \{_L^C\mathbf{T}_1, ..., _L^C\mathbf{T}_n\} \in SE(3)$ as the set of camera pose, and $_L^C\mathbf{T}_i = (_L^C\mathbf{R}_i, _L^C\mathbf{t}_i)$ as the i-th coarse camera pose $(i = 1, ..., n)$. Note that the world frame, denoted as W, aligns with the LiDAR-based point cloud frame L, i.e., $(_L^C\mathbf{R}_i, _L^C\mathbf{t}_i) = (_W^C\mathbf{R}_i, _W^C\mathbf{t}_i)$. These color images are used in the subsequent stages of the pipeline. To provide a comprehensive explanation, the rest of the initialization stage's components (Section III-B) are individually described in detail. We begin Section III-B.1 with the method of point cloud adaptive voxelization and hidden points removal. Subsequently, Section III-B.2 introduces the point cloud co-visibility estimation method. Section III-C demonstrates the effectiveness of our proposed approach in globally optimizing camera poses for all panorama images.

B. Initialization

1) Adaptive Voxelization in Hidden Points Removal: The primary objective of this component is to identify the visible portion of the point cloud from a given viewpoint. [5] involves applying the Hidden Point Removal (HPR) [29] for this operation. This operator involves two essential steps: point transformation and construction of a convex hull. However, the existing 3D convex hull algorithms are not well-suited to the parallel computing model of GPUs [30], which makes them unsuitable for processing large-scale point clouds [12]. To address this challenge, we enhance operational efficiency by leveraging the root nodes within the voxel map to predefine a maximum search distance, while the leaf nodes serve as a representation of the point cloud. This

optimization significantly expedites the computation of the convex hull. However, it is essential to acknowledge that the utilization of fixed-resolution voxel maps entails a trade-off between segmentation time and accuracy. To better adapt to the environment, we have integrated an adaptive voxelization method [16], [25], [26] into our workflow to accelerate the process further.

(a) Point Cloud Input   (b) Visible Voxels of Point Cloud

(c) Panorama Image of Visible Points   (d) Panoramic View of Visible Voxels
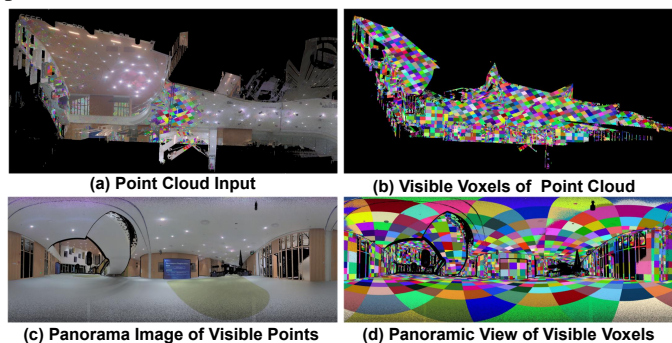
Fig. 3. Adaptive voxelization in Hidden Points Removal. The input point cloud is organized using an adaptive voxel-based data structure for accelerated processing. After adaptive voxelization, each voxel represents a flat surface, maintaining a consistent visible relationship.

2) Point Cloud Co-visibility Estimation: We begin by outlining the pipeline for establishing a point cloud co-visibility estimation method, which aims to reduce the impact of point cloud surface noise on visibility relationships. To achieve this, we first subdivide the global point cloud into multiple voxels and assess the visibility of each point to determine the visible region for each keyframe I_i based on the camera view. This operation is illustrated in Fig. 3 (a-d). The set of visible point clouds is denoted as $\mathbf{P}^V = P_1, .., P_n \subset^L \mathbf{P}$, where $P_i = p_1^{idx_1}, ..., p_m^{idx_m}$ represents the set of visible points for keyframe I_i, idx_i denotes the index of the voxel node that the point p_i belongs to, and m is the count of visible points. Next, we construct a co-visibility graph based on the co-visibility of the point cloud. If the count of visible points in two keyframes sharing the same voxel indexes exceeds a predefined threshold (e.g., half of the minimum count of visible points among these keyframes), an edge is established between them. The co-visible points are added to each keyframe's set of visible points, denoted as $P_i \rightarrow P_i^+$, where P_i^+ represents the updated visible points set for each keyframe. The set of co-visible points is denoted as $\mathbf{P}^{co} = \{P_i^+ \cap P_j^+\}_{i \neq j}$. As shown in Fig. 4, we provide

an example of the method's results mentioned above in a 2D space, offering a quantitative comparison between our approach and the default intersection method.
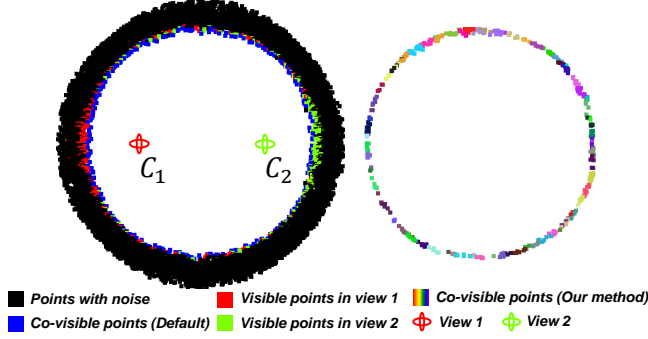

Fig. 4. An illustration of point cloud co-visibility estimation on a 2D circle simulated point cloud data.

C. Camera Pose Optimization

1) Loss Function: Given the co-visible points set $\boldsymbol{P}^{co} = \left\{ P_i^+ \cap P_j^+ \right\}_{i \neq j}$ and a sequence of keyframes $\mathcal{I}$, our objective is to find the optimal camera poses $_L^C\mathcal{T}$ at which panorama images are taken, where L and C are the LiDAR-based point cloud and camera frames, respectively. $_L^C\mathbf{T}_i \in SE(3)$ denotes the extrinsic parameters from the LiDAR-based point cloud frame to the camera frame. Specifically, Fig. 5 demonstrates the LiDAR-based point cloud and camera coordinate systems, as well as the projection function $\Pi(\cdot) : \mathbf{R}^3 \to \mathbf{R}^2$ that maps a 3D point $^C\mathbf{p}^{co} = (x, y, z)$ to a point $\mathbf{p} = (u, v) = (\varphi, \theta)$ in the panorama images' coordinate frame, where $(\boldsymbol{u}, \boldsymbol{v}) \in [0, H) \times [0, W)$ and $(\varphi, \theta) \in (\frac{-\pi}{2}, \frac{\pi}{2}) \times [-\pi, \pi]$ denote the coordinates within the image plane and sphere space. $^C\mathbf{p}^{co} =_L^C \mathbf{T}_i\mathbf{p}^{co}$ project the point cloud from point cloud frame into the camera frame. This function could be explicitly written as follows:

$$\Pi(x, y, z) = \left(\frac{H}{\pi} \arctan(\varphi), \frac{W}{2\pi} \arctan(\theta) \right),$$
$$(\varphi, \theta) = \left(\frac{z}{\sqrt{x^2 + y^2}}, \frac{y}{x} \right). \quad (1)$$

Furthermore, let $\Gamma_i(\mathbf{p}, I_i)$ indicate the function that maps the 2D coordinates $\mathbf{p}$ to pixel values from the keyframes $I_i \in \mathcal{I}_\mathbf{p}$, where $\mathcal{I}_\mathbf{p}$ denotes the set of images associated with $\mathbf{p}$. Under this setup, $\Gamma_i\left(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right), I_i\right)$ be the color at the image coordinates of the projection of $\boldsymbol{p}^{co}$ onto I_i, given an extrinsic matrix $_L^C\mathbf{T}_i$. We want to maximize the agreement within $\left\{\Gamma_i\left(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right), I_i\right)\right\}_{I_i \in \mathcal{I}_\mathbf{p}}$ for each co-visible point $\mathbf{p}^{co}$. To this end, we introduce an auxiliary variable $C(\mathbf{p}^{co})$ to represent the color of $\mathbf{p}^{co}$. If the co-visible point $\mathbf{p}^{co}$ is perfectly aligned with the keyframes $\mathbf{I}$, one could except the projected pixel values' set $\left\{\Gamma_i\left(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right), I_i\right)\right\}_{I_i \in \mathcal{I}_\mathbf{p}}$ to be very close to the point color values $C(\mathbf{p}^{co})$. Our goal is to optimize the set of camera pose $_L^C\mathcal{T}$, with the set of the auxiliary variable $\mathbf{C} = \{C(\mathbf{p}^{co})\}$, where the objective is to minimize the discrepancy between $\left\{\Gamma_i\left(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right), I_i\right)\right\}_{I_i \in \mathcal{I}_\mathbf{p}}$

and $C(\mathbf{p}^{co})$. Let $\mathbf{P}_i^{co} = P_i^+ \cap \{P_j^+\}_{j \neq i}$, where $\mathbf{P}_i^{co}$ denotes the co-visible points set from keyframes I_i. This can be formulated as follows:

$$E(\mathbf{C},_L^C\mathcal{T}) = \sum_i \sum_{\mathbf{p}^{co} \in \mathbf{P}_i^{co}} \|\Gamma_i(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right)) - C(\mathbf{p}^{co})\|_2. \quad (2)$$

The loss function from Equation 2. is the point cloud-centric sampling loss, which is evaluated at the projected location of every co-visible point in the point cloud. This sampling loss has several advantages. First, It fairly incorporates all points in the point cloud, thus making it overcome the severe visual distortion of omnidirectional images. In addition, the $\mathbf{P}_i^{co}$ for each image I_i is definitely stable for 360-degree cameras. Thus, it allows us to bypass the non-differentiable changes in the visibility relationship between the camera and point cloud during optimization, significantly reducing computational costs.
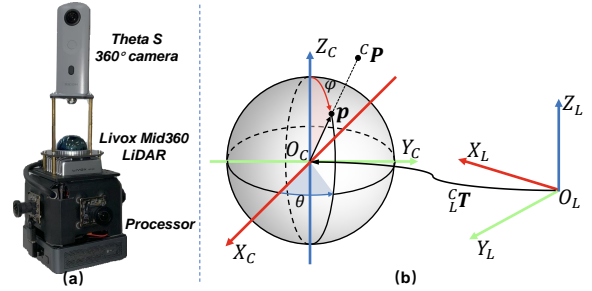

Fig. 5. (a) Our mobile mapping system and (b) LiDAR-camera coordinate system.

2) Alternating Optimization: We have implemented an alternating optimization scheme, inspired by [22], to minimize the loss function. The basic idea is to alternate optimizing between $_L^C\mathcal{T}$ and $\mathbf{C}$. When optimizing $\mathbf{C}$, $_L^C\mathcal{T}$ is kept fixed, and vice versa. To initialize the camera poses when $_L^C\mathcal{T}$ is fixed, we use the extrinsic calibration between the camera and the LiDAR Odometry. We then colorize the point cloud based on the initial coarse camera poses and keyframe images. However, due to slight inaccuracies in the point cloud map, illumination changes from different camera views, and coarse camera poses, colorization errors are inevitable. To mitigate this problem, we use a form of robust average. Each point in the point cloud has a set of candidate colors, and the number of the candidate colors is denoted as $\mathbf{k}$. Then, the nonlinear problem Equation 2 is transformed into a linear least-squares problem with a closed-form solution.

$$C(\mathbf{p}^{co}) = \frac{1}{\mathbf{k}_{\mathbf{p}^{co}}} \sum_{I_i \in \mathcal{I}_{\mathbf{p}^{co}}} \Gamma_i(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right)). \quad (3)$$

When $\mathbf{C}$ is fixed, the loss function Equation 2. decomposes into an independent loss for each $_L^C\mathbf{T}_i$:

$$E_i(\mathbf{T}) = \sum_{\mathbf{p}^{co} \in \mathbf{P}_i^{co}} \left\|\Gamma_i(\Pi\left(_L^C\mathbf{T}_i\mathbf{p}^{co}\right)) - C(\mathbf{p}^{co})\right\|_2. \quad (4)$$

The core part of OmniColor consists of a simple gradient descent on the sampling loss Equation 4, which is computationally efficient. Meanwhile, the loss function can

effectively handle the comprehensive analysis of the 360-degree image and demonstrate robustness against visual distortion. Fig. 6 shows an example of our loss function optimization process on the simulation dataset.

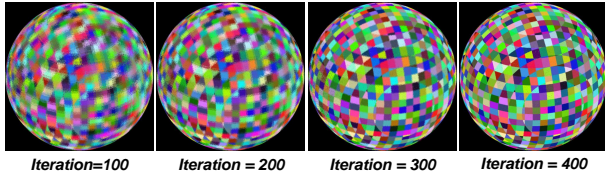| Iteration=100 | Iteration = 200 | Iteration = 300 | Iteration = 400 |

Fig. 6. The colorization result during optimization process. Our method is capable of processing sphere simulation point clouds, which inherently lack distinct geometric features.

IV. EXPERIMENTS

To evaluate the performance of the proposed method, we compare it against other state-of-the-art approaches using two categories of datasets. The first includes data collected from a mobile mapping device, specifically the Livox Mid360 solid-state LiDAR for high-resolution point cloud measurements and a RICOH Theta S 360-degree camera for qualitative analysis (see Fig. 7). The second category comprises data captured from a stationary LiDAR Scanner named BKL360 within the HKUST Guangzhou campus premises, including four scenes with over 10 station frames from various positions and orientations. Additionally, we utilize the publicly available Omniscenes dataset [31], consisting of panorama image sequences with ground truth camera poses and scene 3D point clouds, for comparison with the SfM-based method [32]. Finally, a simulated dataset is employed for an ablation study on our method regarding point cloud co-visibility estimation.

To quantitatively compare our optimized parameters with ground-truth values in terms of rotation error (unit: degree) and translation error (unit: centimeters), random rotational and translational noise is added to the original camera views before running the optimization algorithm. The initial rotation and translation errors are computed as $5.0°/10cm$. The following presentation of experimental results aims to directly demonstrate the efficacy of our method through quantitative and qualitative analysis.

TABLE I
ROTATION ERRORS (DEGREES) AND TRANSLATION ERRORS
(CENTIMETERS) ON HKUST GUANGZHOU CAMPUS DATASET

	Scene 1	Scene 2	Scene 3	Scene 4	Average
Depth Edge-Based [14]	0.89/5.45	1.52/6.75	0.76/4.78	3.67/5.81	1.71/5.70
Intensity Edge-Based [15]	0.82/4.53	0.96/3.67	1.98/6.83	4.37/7.21	2.03/5.56
Ours	**0.05/3.83**	**0.03/2.56**	**0.04/2.48**	**0.07/3.35**	**0.475/3.06**

*The initial rotation and translation errors are 5° /10 cm.

A. Quantitative Results

Our method is specifically tailored for the utilization of a 360-degree camera. To facilitate comparison with other methods, we rectify the panorama image into a pinhole image encompassing FOV of 160 degrees and subdivide the input point cloud into multiple local maps centered around coarse camera views. Specifically, we compare our methods with two types of edge feature-based extrinsic calibration approaches: depth-continuous edges [14] in the point cloud and edge features extracted from LiDAR intensity images [15], which align the local maps with color images through extrinsic calibration. We have fine-tuned the parameters of each method to achieve the best performance through the authors' efforts. Consistent parameter settings are applied across all scenes for each method.

The experimental results are summarized in Table I. Our method outperforms other approaches in terms of accuracy across all scenes in the HKUST Guangzhou campus dataset. In contrast to edge feature-based methods, which exhibit sensitivity to the environment and encounter failures in several local maps, our approach demonstrates robustness in handling diverse mapping scenes. We acknowledge that there were failure cases with the edge feature-based method; therefore, for calculating the final errors of edge feature-based methods presented in Table I, we only evaluate successfully aligned local maps. However, even within these successful scenes, our method consistently achieves superior accuracy. The improved performance may be attributed to two reasons. Firstly, the color space of the omnidirectional camera is more sensitive than the edge features proposed in [14], [15]. In our specific scenario, the initial noise is minimal, resulting in coarse camera poses that are close to the global minimum solution where feature-based methods' gradient approaches zero. However, our direct method still possesses a clear optimized direction leading to more precise outcomes. Additionally, it's challenging to identify reliable correlation features from various modalities, which can cause edge feature mismatches and degrade alignment accuracy.

We further evaluate with the SfM method [32], which jointly optimizes image poses, feature points, and calibration parameters using LiDAR-SLAM as priors. We conduct experiments on the Omniscenes dataset. This dataset contains panorama videos with ground truth poses and 3D point clouds, which can be regarded as the data captured by a mobile mapping system. For a fair comparison, we convert the panorama image to six cube pinhole images and initialize the coarse camera poses by randomly perturbing the ground-truth values ($5.0°$ for rotation; $10cm$ for translation). In the optimization processing, we add rigid constraints on the six cube pinhole images derived from the same panorama image. The results are shown in Table II. We notice that the accuracy of our approach is close to the joint SfM method [32]. Although achieving approximate accuracy, ours is a kind of direct method that does not need any feature extraction and matching process that can significantly reduce computational costs. In addition, our optimization scheme makes it easy to exploit modern GPU power to update all poses and point cloud colorization results in parallel.

B. Qualitative Results

In addition to the qualitative comparison, we have specifically chosen various failure scenarios involving feature-based methods from different sequences within the HKUST
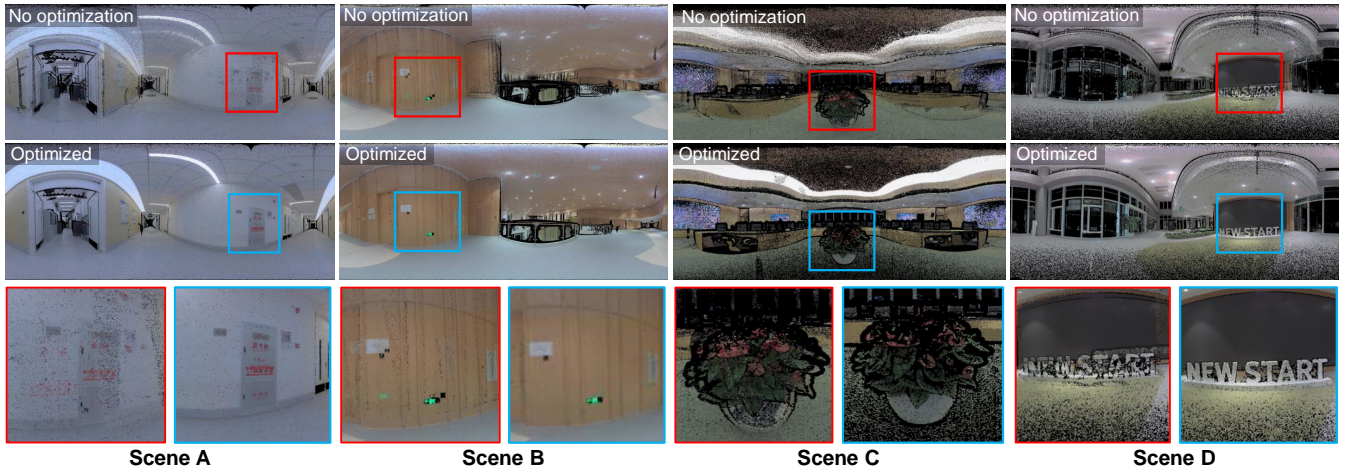
Fig. 7. The rendered images of the colorization result of the HKUST Guangzhou campus dataset (Scene A-D). Left-red: The colored point with noise before optimization. Right-blue: The colored point cloud optimized by our method.

TABLE II
ROTATION ERRORS (DEGREES) AND TRANSLATION ERRORS (CENTIMETERS) ON OMNISCENES DATASET

	PyebaekRoom	Room 1	Room 2	Room 3	Room 4	Room 5	WeddingHall 1	Average
Prior Pose-based SfM [32]	0.26/2.75	0.34/4.25	0.26/3.46	0.35/3.57	0.43/**3.42**	0.35/4.53	**0.21/3.26**	0.31/3.61
Ours	**0.12/2.54**	**0.24/3.42**	**0.18/2.96**	**0.34**/3.42	**0.38**/4.28	**0.28/3.53**	0.24/4.75	**0.25/3.56**

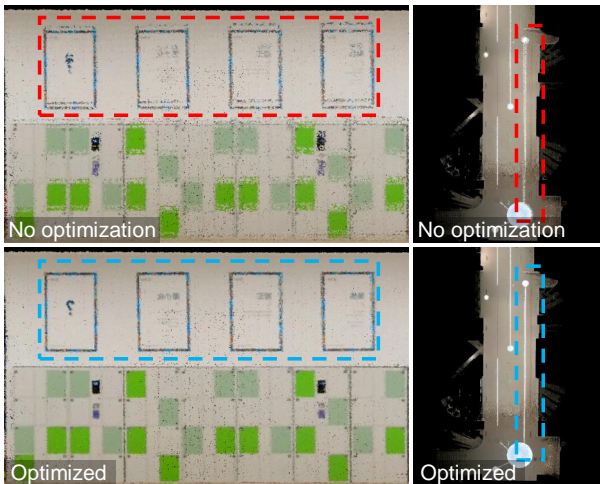*The initial rotation and translation errors are 5° /10 cm.

Fig. 8. The colored point cloud reconstructed by our mobile mapping device. Top: Colored based on interpolated camera poses. Bottom: colored based on optimized camera poses

Guangzhou campus dataset. These scenarios highlight the consistent high-quality results achievable with our method across all scenes. Fig. 7 illustrates how our method significantly enhances point cloud colorization outcomes. In the case of the mobile mapping dataset, we conducted a comparative analysis against [5], as shown in Fig. 8. The results clearly indicate the substantial improvement in colorization achieved by our method.

C. Ablation Study on Point Cloud Co-visibility Estimation

We perform this experiment using a simulated dataset that we generated, consisting of points within a sphere with a radius of 10m. Fig. 6 illustrates the progress of our colorization process during optimization. In this section, we conduct a quantitative analysis to assess the impact of point

noise on the optimization results. Fig. 9 introduces noise ranging from 1cm to 10cm. The results demonstrate that our method effectively mitigates the noise impact on the point cloud's surface, leading to more precise camera poses.
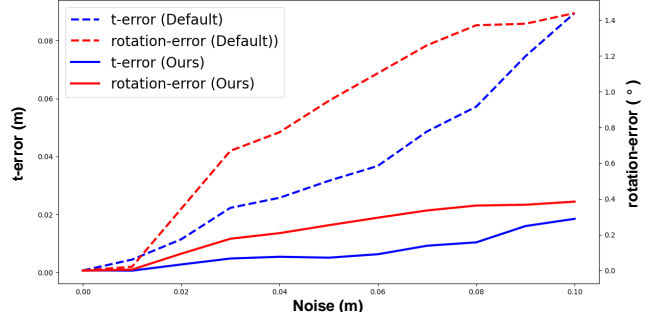

Fig. 9. Ablation study about the points co-visibility estimation method for camera pose optimization.

V. CONCLUSIONS

In this paper, we propose a novel and efficient algorithm (OmniColor) to colorize point clouds using an independent 360-degree camera. It takes advantage of the omnidirectional perception view of the camera and eliminates the impact of severe spherical distortion while maintaining computational efficiency. In experiments conducted on both our proprietary dataset and the public dataset, OmniColor outperforms existing algorithms in both accuracy and stability compared to SOTA methods. Moreover, OmniColor seamlessly integrates with any mobile mapping system, opening up a world of possibilities for its application in various domains, including virtual reality and robotics, where the need for clear and precise colored point cloud maps is paramount.

REFERENCES

[1] E. Nocerino, F. Poiesi, A. Locher, Y. T. Tefera, F. Remondino, P. Chippendale, and L. Van Gool, "3d reconstruction with a collaborative approach based on smartphones and a cloud-based server," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. W8, pp. 187–194, 2017.

[2] R. Hussain, M. Pizzo, G. Ballestin, M. Chessa, and F. Solari, "Experimental validation of photogrammetry based 3d reconstruction software," in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*. IEEE, 2022, pp. 1–6.

[3] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[4] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[5] P. Vechersky, M. Cox, P. Borges, and T. Lowe, "Colourising point clouds using independent cameras," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3575–3582, 2018.

[6] J. Lin and F. Zhang, "R^3 LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 672–10 678.

[7] L. Jiarong and Z. Fu, "R^3 LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator," *arXiv preprint arXiv:2209.03666*, 2022.

[8] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4003–4009.

[9] W. Zhen, Y. Hu, J. Liu, and S. Scherer, "A joint optimization approach of lidar-camera fusion for accurate dense 3-d reconstructions," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3585–3592, 2019.

[10] S. Sumikura, M. Shibuya, and K. Sakurada, "Openvslam: A versatile visual slam framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2292–2295.

[11] H. Huang and S.-K. Yeung, "360vo: Visual odometry using a single 360 camera," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5594–5600.

[12] J. Cui and S. Schwertfeger, "Cp+: Camera poses augmentation with large-scale lidar maps," in *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2022, pp. 69–74.

[13] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[14] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[15] Z. Miao, B. He, W. Xie, W. Zhao, X. Huang, J. Bai, and X. Hong, "Coarse-to-fine hybrid 3d mapping system with co-calibrated omnidirectional camera and non-repetitive lidar," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1778–1785, 2023.

[16] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small fov lidars and cameras using adaptive voxelization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[17] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 5692–5698.

[18] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R 2 live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[19] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.

[20] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, "Dv-loam: Direct visual lidar odometry and mapping," *Remote Sensing*, vol. 13, no. 16, p. 3340, 2021.

[21] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[22] Q.-Y. Zhou and V. Koltun, "Color map optimization for 3d reconstruction with consumer depth cameras," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, pp. 1–10, 2014.

[23] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[24] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[25] Z. Liu, X. Liu, and F. Zhang, "Efficient and consistent bundle adjustment on lidar point clouds," *arXiv preprint arXiv:2209.08854*, 2022.

[26] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8518–8525, 2022.

[27] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[28] E. Ong, W. Lin, Z. Lu, S. Yao, F. Pan, L. Jiang, and F. Moschetti, "A no-reference quality metric for measuring image blur," in *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, vol. 1. Ieee, 2003, pp. 469–472.

[29] S. Katz and A. Tal, "On the visibility of point clouds," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1350–1358.

[30] R. M. e Silva, C. Esperança, and A. Oliveira, "Efficient hpr-based rendering of point clouds," in *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2012, pp. 126–133.

[31] J. Kim, C. Choi, H. Jang, and Y. M. Kim, "Piccolo: point cloud-centric omnidirectional localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3313–3323.

[32] D. Lee, S. Ryu, S. Yeon, Y. Lee, D. Kim, C. Han, Y. Cabon, P. Weinzaepfel, N. Guérin, G. Csurka *et al.*, "Large-scale localization datasets in crowded indoor spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3227–3236.