2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Robust 360-8PA: Redesigning The Normalized 8-point Algorithm for 360-FoV Images

Bolivar Solarte[1], Chin-Hsuan Wu[1], Kuan-Wei Lu[1], Yi-Hsuan Tsai[3], Wei-Chen Chiu[2], Min Sun[1]

***Abstract*— In this paper, we present a novel preconditioning strategy for the classic 8-point algorithm (8-PA) for estimating an essential matrix from 360-FoV images (i.e., equirectangular images) in spherical projection. To alleviate the effect of uneven key-feature distributions and outlier correspondences, which can potentially decrease the accuracy of an essential matrix, our method optimizes a non-rigid transformation to deform a spherical camera into a new spatial domain, defining a new constraint and a more robust and accurate solution for an essential matrix. Through several experiments using random synthetic points, 360-FoV, and fish-eye images, we demonstrate that our normalization can increase the camera pose accuracy about 20% without significantly overhead the computation time. In addition, we present further benefits of our method through both a constant weighted least-square optimization that improves further the well known Gold Standard Method (GSM) (i.e., the non-linear optimization by using epipolar errors); and a relaxation of the number of RANSAC iterations, both showing that our normalization outcomes a more reliable, robust, and accurate solution.**

I. Introduction

Estimating the relative pose between different views of the same scene has been studied for decades in Computer Vision and Robotics. For instances, Visual Odometry (VO), Simultaneous Localization and Mapping (SLAM), Structure from Motion (SFM), among others, generally leverage this primary estimation for initializing the first camera poses, triangulate landmarks in 3D, re-localize the camera pose, and prune out outliers correspondences from the system.

In practice, the goal of estimating a camera pose between two images relies on finding a geometry constraint for their pixels. This is known as the essential matrix or epipolar constraint. In general, the procedure of calculating an essential matrix includes two main steps: 1) An abundant number of salient pixels, i.e., *key-features*, are extracted from each image, followed by matching them across different views; 2) Based on that correspondences, the essential matrix is calculated satisfying the epipolar constraints. Finally, after the essential matrix is derived, the relative camera pose can be recovered by singular value decomposition [1], [2].

Several algorithms have been proposed to find an essential matrix, however the most widely used solutions are the five-point (5-PA) [4] and eight-point (8-PA) [3] algorithms. Despite the former uses the minimum number of correspondence points needed for calibrated cameras, its implementation usually relies on a polynomial approximation with multiple solutions. In contrast, the 8-PA is a linear method without the ambiguity in its outcome. In general, the 8-PA is mostly used for 360-FoV images (e.g., [5], [6], [7], [8], [9], [10]), due to its simplicity and proved stability under large field of views [11]. On the other hand, unlike the 5-PA, the 8-PA requires more iterations for outlier removal using a RANSAC evaluation, hence increasing its computation time for a large ratio of outliers [12]. This defines a clear disadvantage of the 8-PA.

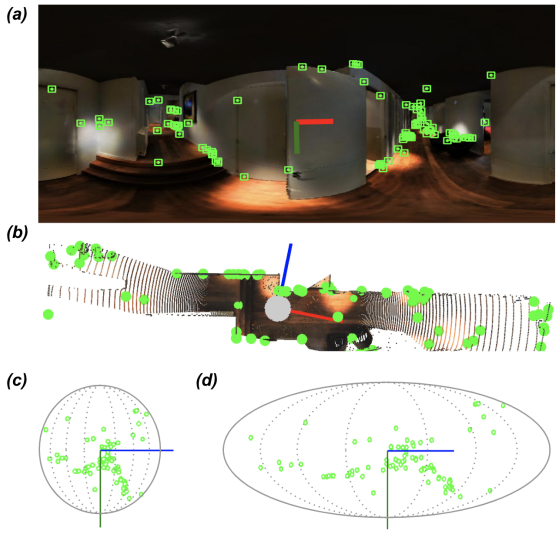

Fig. 1. **Illustration for the distribution of key-features in a 360-FoV image.** (a) and (b) present a 360-FoV image and its top view (point cloud) respectively, with its key-features denoted as landmarks in green color. In (c), the same landmarks presented in (a) and (b) are projected into a unit sphere. In (d), our proposed normalization scheme (cf. Section IV-A) has been applied to the spherical features in (c). Note that the key-features in the ovoid surface (d) are geometrically shifted compared to (c) due to our normalization. As a result, (d) expands the relative angles between every key-feature, which in turn leads to a more stable solution when using the 8-PA method [3].

Although studies in [11] show that a wider FoV may increase the stability of the 8-PA for spherical cameras, these assume that matched key-points in the image are well-distributed in the whole FoV. However, in practice, that distribution mainly depends on external factors, which sometimes yields to clustered or uneven distribution of key-points (see Fig. 1) (e.g. [10]).

In this work, we improve the 8-PA [3] for spherical cameras by re-defining *the pre-conditioning* strategy proposed by [13] to be applied to spherical projection, which effectively deals with outliers and uneven key-feature distributions for 360-FoV images. Additionally, we extend the usage of our novel pre-conditioning by proposing a constant weighted least-square solution, which improves the Gold Standard Method (GSM) [14], [2] that is usually used to refine the

[1] National Tsing Hua University
[2] National Chiao Tung University
[3] NEC Labs America

camera pose. Lastly, we also present results comparing our solution under a RANSAC evaluation, showing that our preconditioning is capable to effectively deal with outliers, hence potentially reducing the number of required iterations.

We evaluate our methods under both sequences of real 360-FoV and fish-eye scenarios, where the former is our own dataset, collected from Matterport3D [15] and rendered using MINOS [16], while the latter is the TUM-VI dataset [17]. We show that our method significantly outperforms the state-of-the-art 8-PA for spherical cameras without the overhead in computation time, demonstrating the robustness of our method against noise and outliers. In favor of the research community, the source code is available at `https://github.com/EnriqueSolarte/robust_360_8PA`.

II. Related Work

Several approaches have been developed to estimate an essential matrix based on matched key-features. Nevertheless, the most well-known approaches are still the 5-PA [4] and 8-PA [3], where the former has been improved largely in past years, e.g., [18], [19], [20], [21]. However, all of them rely on a polynomial approximation, which inevitably leads to multiple essential matrix solutions. In contrast, the 8-PA [3] is a simpler and linear method that always outputs a unique result.

In [13], a normalization strategy is introduced upon 8-PA [3], improving homogeneity in the input data, and robustness against noise. However, this preconditioning was mainly designed for uncalibrated pinhole cameras. Later, in [22], the normalization [13] is further explained in terms of a generalized total least square problem, which opens the idea of a general normalization. However, in this work, the authors focus only on perspective cameras, proving that the normalization [13] (i.e., an isotropic and non-isotropic normalization), indeed, reduces the effects of noise in the solution of the 8-PA for key-point described in a homogeneous plane. In the literature there is no evidence for normalization for spherical projection.

For spherical projection, the 8-PA has been widely used as a standard solution (e.g., [23], [8], [9], [14], [24], [5]), that estimates a quick initial guess, where other methods can efficiently refine it further [1], [2], [23]. Note that a reliable and fast estimation of an initial guess is always desired.

Several works have been proposed to explain the stability of the 8-PA as a linear least-square problem, e.g., [25], [22]. However, most of them assume a known noise distribution in the data. Instead, [11] recently demonstrates that without any noise assumption, the 8-PA stability is highly related to the FoV of the used image. However, the statement assumes a uniform distribution of key-features along the FoV.

In general, a key-feature distribution is highly defined by external factors, such as poor scene illumination, lack of texture, among others [2], [23], [26]. Indeed, for spherical cameras, due to the high distortion, uneven feature distributions are a critical issue for matching correspondences as studied in [10]. Therefore, using key-features directly from 360-FoV images may define an uneven distribution.

III. Preliminaries

A. Spherical Projection and Bearing Vectors

Unlike perspective images where a homogeneous plane is used, 360-FoV images are generally represented by a centralized spherical projection, which can be described as:

$$\begin{bmatrix}\theta\\ \phi\end{bmatrix} = \begin{bmatrix}2\pi/W & 0 & -\pi\\ 0 & -\pi/H & \pi/2\end{bmatrix}\begin{bmatrix}u\\ v\\ 1\end{bmatrix} \tag{1}$$

$$\mathbf{q_n} = \begin{bmatrix}x_n\\ y_n\\ z_n\end{bmatrix} = \begin{bmatrix}cos(\phi)sin(\theta)\\ -sin(\phi)\\ cos(\phi)cos(\theta)\end{bmatrix}, \tag{2}$$

where (1) projects a pixel (u,v) into the spherical coordinate (θ,ϕ), and then (2) transforms that coordinate into a unit vector $\mathbf{q_n}$. Hereinafter, we name $\mathbf{q_n}$ as a *bearing vector*. Note that W and H are the width and height of the 360-FoV image.

B. Epipolar Constraint and the Eight-Point Algorithm

Without loss of generality, the same epipolar constraint defined for perspective images can be applied to spherical projection [1]. Therefore, given a pair of 360-FoV images, two unit sphere cameras can be projected (i.e., C_1 and C_2 in Fig. 2-(d)), from which the *bearing vectors* $\mathbf{q_1}$ and $\mathbf{q_2}$ can be also defined. Then, the epipolar constraint, which defines the coplanarity of $\mathbf{q_1}$ and $\mathbf{q_2}$ onto an epipolar plane π, can be described as follows:

$$\mathbf{q_2}^\top \mathbf{E}\mathbf{q_1} = \mathbf{0}, \text{ with } \mathbf{E} = [\mathbf{t}\,]_\times \mathbf{R}\ , \tag{3}$$

where $\mathbf{E}$ represents the essential matrix $\in \mathbb{R}^{3\times3}$ with rank of 2; $[\mathbf{t}\,]_\times$ stands for a skew-symmetric matrix coming from $\mathbf{t}\in\mathbb{R}^3$; and $\mathbf{R}$ defines a relative camera rotation $\in SO(3)$.

We can then reformulate (3) into a *Total Least Square Problem* as follows:

$$\mathbf{A}[\mathbf{E}]_\mathrm{v} = 0\ , \tag{4}$$

where $[\mathbf{E}]_\mathrm{v}$ is a vector of the coefficients in matrix $\mathbf{E}$ by a row-wise concatenation; $\mathbf{A}$ is an $n\times9$ matrix which is built upon stacking at least $n\geq8$ correspondences of bearing vectors (usually called *observation matrix* [2]), where an i^{th} row of this matrix can be defined by the Kronecker product of two correspondence bearing vectors as $\mathbf{A_i} = \mathbf{q_1^i}\otimes\mathbf{q_2^i}$.

Afterward, Singular Value Decomposition (SVD) is applied to $\mathbf{A}$ for finding the solution of (4), in which the last column of the orthogonal subspace of $\mathbf{A}$ defines $[\mathbf{E}]_\mathrm{v}$, and can then be reshaped into $\mathbb{R}^{3\times3}$ and forced into rank(2). This procedure is called *Direct Linear Transformation* (DLT) [2].

IV. Robust Spherical Normalization

As the key-features are primarily located at areas with high texture, a wider FoV image cannot always provide an uniformly distributed locations of key-features. Thus, we propose our normalization strategy for the classic 8-PA to reduce the effect when the key-features are not uniformly distributed.

First, we derive the mechanism introduced in [13] but with our normalization applied, and then we define a new vector space for stabler DLT solution of an essential matrix. We also discuss the reasons why our normalization increases

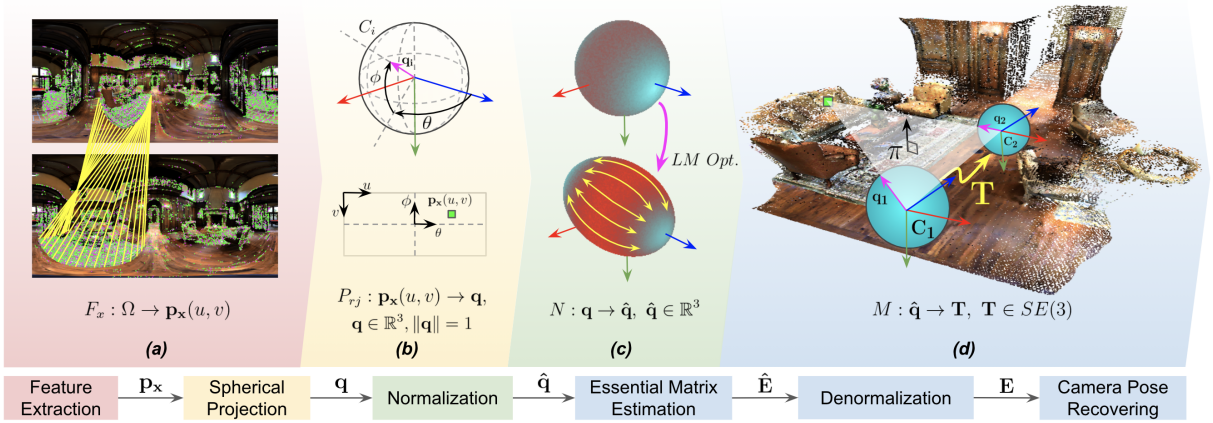

Fig. 2. **Normalized 8-PA for spherical cameras.** Our method takes two raw 360-FoV images as input, from where we extract several key-features and matching their correspondences across the images as shown in (a). Next, in (b), the same landmarks are projected into a unit sphere by spherical projection (cf. III-A). Afterward, the sphere is deformed by our normalization method (cf. IV) in order to make a better distribution of key-features (c). Lastly, we estimate the essential matrix by using DLT over the normalized domain (cf. IV-A).

the stability of the 8-PA. Lastly, we present two non-linear optimizations aiming to improve both the 8-PA and GSM methods without overhead computation time.

A. Normalization

Unlike [13], we define our normalization as a transformation that deforms bearing vectors from an unit sphere camera into a ovoid surfaces (see Fig. 3-(c)). This matrix transformation is defined as follows:

$$\mathbf{N} = \begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & K \end{bmatrix} , \tag{5}$$

$$\hat{\mathbf{q}}_i = \mathbf{N}\ \mathbf{q}_i \ , \tag{6}$$

where $S, K \in \mathbb{R}$ and $|\mathbf{N}| \neq 0$. For convenience, our normalization is designed as a diagonal matrix $\mathbb{R}^{3\times3}$, controlled by two parameters, S and K, as presented in (5), which allows us to deform bearing vectors along XY and Z directions, respectively. Thus, we represent the normalization (6) with the epipolar constraint presented in (3):

$$\hat{\mathbf{q}}_2^\top\ \mathbf{N}_2^{-\top}\ \mathbf{E}\ \mathbf{N}_1^{-1}\ \hat{\mathbf{q}}_1 = 0 \ . \tag{7}$$

By further arranging (7), we can build our normalized constraint as follows:

$$\hat{\mathbf{q}}_2^\top \hat{\mathbf{E}}\ \hat{\mathbf{q}}_1 = 0 \ , \tag{8}$$

$$\mathbf{E} = \mathbf{N}^\top \hat{\mathbf{E}}\ \mathbf{N} \ . \tag{9}$$

Note that (8) is embodied by $\hat{\mathbf{E}}$, which stands for our normalized essential matrix in the normalized domain; thus, we can find $\hat{\mathbf{E}}$ by using the standard DLT procedure described in Sec. III. In this way, we can recover the original essential matrix $\mathbf{E}$ by left-right multiplying the matrix $\mathbf{N}$ as described in (9), which we call *denormalization*. Finally, the relative camera pose $\mathbf{T} \in SE(3)$ can be recovered from $\mathbf{E}$ by SVD. To be clear, we present the procedure in Fig. 2.

B. Deformation of Spherical Projection

Intuitively, normalizing bearing vectors through a non-rigid transformation (5) deforms the unit sphere into a different spatial domain which in turns defines a new observation matrix $\mathbf{A}$. To explain why this normalization leads to more stable DLT solution for (8), we present two properties that define stability for 8-PA in the context of spherical cameras.

First, based on [11], we assert that the FoV of an image has a strong correlation with the stability of an essential matrix estimation when using the 8-PA. The main reason is because large FoV images are prone to define large internal angles between its bearing vectors, i.e. the angles β_{ij}^1 and β_{ij}^2 in Fig. 3(a). This can be mathematically justified by representing these internal angles in terms of the observation matrix $\mathbf{A}$ as shown in (10), which in turn evaluates a bound for the second least singular value of $\mathbf{A}$ (i.e. σ_8) as presented in (11). For more details, we refer to Sec. 3.4. in [11]:

$$\left\|\mathbf{A}\mathbf{A}^\top\right\|_F^2 = \sum_i \sum_j (cos_i^2 \beta_j^1)(cos_i^2 \beta_j^2) \ , \tag{10}$$

$$\sigma_8 \leq \sqrt{\frac{n}{8} - \frac{1}{8}\sqrt{\frac{8\left\|\mathbf{A}\mathbf{A}^\top\right\|_F^2 - n^2}{7}}} \ . \tag{11}$$

Based on [11], [25], the error in an essential matrix $\mathbf{E}$ can be defined as a function of σ_8 as follows:

$$\Delta\mathbf{E} \leq \frac{\|\mathbf{Q}\|_2}{\sigma_8} \ , \tag{12}$$

where $\Delta\mathbf{E}$ is the error in the essential matrix, and $\mathbf{Q}$ represents the perturbation matrix which embodies the noise and outliers in bearing vectors. In practice, based on (12), we assert that larger internal angles, β_{ij}^1 and β_{ij}^2, can evaluate larger σ_8 values, which in turn result in a lower $\Delta\mathbf{E}$ error. Therefore, if we deform the spherical projection by increasing the internal angles between bearing vectors, we will have larger σ_8 values and thus obtain a more stable solution. In addition, based on the translation vector $\mathbf{t}$ between cameras and the distance to some landmarks in the scene, we can define the motion parallax for the spherical projection as

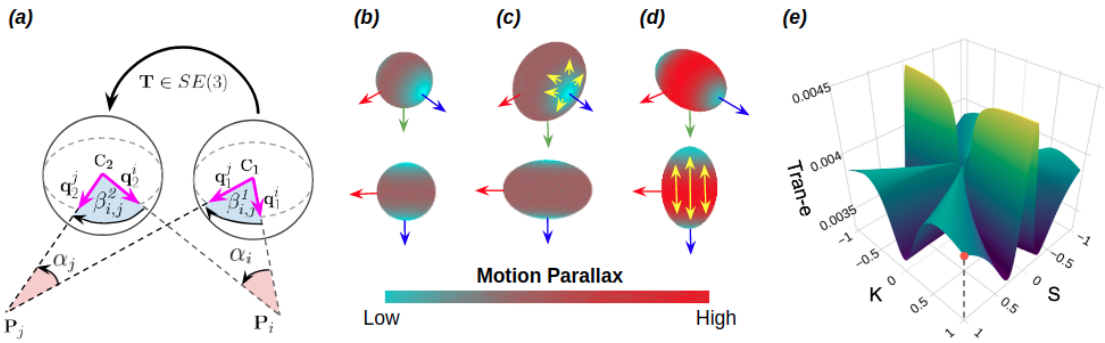

Fig. 3. **Stability of the spherical projection**. In panel (a) given the locations of the landmarks $\mathbf{P}_i$ and $\mathbf{P}_j$ the internal angles β_{ij}^1 and β_{ij}^2 as well as the motion parallax α_i and α_j are defined. In panels (b)-(d), geometric visualizations of our normalization are presented (cf. Sec.IV-A). Panel (b) shows a spherical camera without normalization as a reference. In (c), the effects of setting $S = 2$ are illustrated, while in (d) the effects of $K = 2$. Note that normalizing a spherical camera using $(S,\ K)$ can increase the motion parallax in the data. Lastly, in panel (e) we present a surface loss for translation error build upon several combinations of S and K values from a grid. We can verify that there exists a combination of $(S,\ K)$ that reduces error.

the angles α_j and α_i in Fig. 3(a). Therefore, analogously to the motion parallax defined for perspective projection (e.g. [2], [23]), we can assert that when the angles α_j and α_i are close to zero, the DLT solution of the 8-PA is unable to recover a camera pose. Thus, the lack of motion parallax is one of the degenerative conditions for the 8-PA (e.g. [3], [13], [23]). Here, if we properly dislocate every bearing vector in a particular direction, we can modify the motion parallax and further define a more reliable estimation of a camera pose.

As illustrated in Fig. 3(c), we compute a grid of S and K values defining different normalizing matrices (5). By using random synthetic landmarks in 3D, we apply our proposed normalization to two known spherical cameras as described in Sec. IV-A. Thus, we can visualize that there exists a set of S and K values which improves a camera pose estimation by normalizing bearing vectors.

In the same synthetic environment, we evaluate the motion parallax for every matched bearing vector and plot it in a scale of color in Fig. 3(b)-(d). By deforming the spherical projection along a specific direction, we can magnify the motion parallax in a set of bearing vectors.

C. Non-linear Optimization over S and K

Although the normalized 8-PA proposed by [13] gives us a mechanism to apply our normalization (5) to spherical features, the method does not provide a valid procedure to evaluate a normalized matrix under spherical projection.

Therefore, we propose a non-linear optimization to reduce errors in the epipolar constraint by locally finding an optimal matrix $\mathbf{N}$, parameterized by S and K, which effectively normalizes bearing vectors in spherical projection.

In practice, based on the projected distance in [14], we define our residual error in the epipolar constraint as follows:

$$\varepsilon(\Theta) = \frac{|\mathbf{q}_2^\top\ \mathbf{E}_\Theta\ \mathbf{q}_1|}{\|\mathbf{q}_2\|\ \|\mathbf{E}_\Theta\ \mathbf{q}_1\|}\ , \tag{13}$$

where $\mathbf{E}_\Theta$ represents the essential matrix evaluated by a particular set of Θ parameters. Thus, for our proposed method, Θ represents S and K while as for the optimization of GSM, it is defined as $\xi \in \mathbb{R}^6$, i.e. $\mathbf{T} \in SE(3)$ mapped into the Lie group $\xi \in \mathfrak{se}(3)$ [23]. Thus, we can define our non-linear optimization to find a set of S^* and K^* parameters as follows:

$$S^*, K^* = \underset{S,K\in\mathbb{R}^2}{\operatorname{argmin}} \|\varepsilon(S,K)\|_1^2 . \tag{14}$$

Unlike GSM, which is defined over 6-DoF while minimizing the least-square errors of (13), our proposed optimization (14) is defined over 2-DoF only (i.e. S and K), which do not evaluate a camera pose directly; thus an initial evaluation of the 8-PA is not needed for an initial guess, which in turns does not add the overhead in time. To deploy our optimization (14), we leverage the LM algorithm [27] with both S and K parameters set to 1 as a trivial initial point.

To further show the versatility of our normalization upon the 8-PA solution for 360° images, we also propose a robust constant weighting function to improve the GSM accuracy without increasing its computation time. This optimization is described as follows:

$$\xi^* = \underset{\xi\in\mathbb{R}^6}{\operatorname{argmin}} \sum_i \omega_{iSK}\ \varepsilon_i(\xi)^2\ , \tag{15}$$

where ω_{SK} is a constant vector build upon the residuals $\varepsilon(S^*,K^*)$, which defines the confidence for every matched bearing vector as a probabilistic function $P(\varepsilon(S^*,K^*))$. Combined with our proposed optimization (14), we compute a robust essential matrix $\mathbf{E}_{SK}$ and evaluate a residuals vector $\varepsilon(S^*,K^*)$, from where ω_{SK} is computed as a normal distribution $\mathcal{N}(\varepsilon(S^*,K^*)|\mu,\sigma)$, evaluating μ and σ as the mean and standard deviation of the residuals vector $\varepsilon(S^*,K^*)$, respectively. To deploy the optimization (15), the LM algorithm [27] is also used.

V. Experimental Results

In this section, we conduct experiments to demonstrate that our method achieves a more accurate camera pose estimation for 360-FoV images. We define two environments: synthetic random points (Sec. V-A) and sequence of real-images (Sec. V-B and V-C).

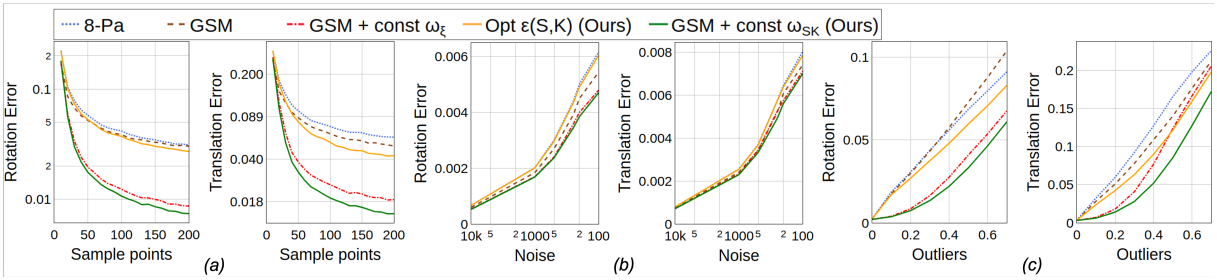

Fig. 4. **Synthetic Points Evaluation.** Unless noted otherwise, all of the experiments described in this figure uses: 200 matched/correspondences bearing vectors in spherical projection as input; von Misses-Fisher noise of $\kappa = 500$ (3.21° of error). In panel (a), a constant outliers ratio of 20% has been added, while in (b), the input data is evaluated as free-outliers data.

For experiments in Sec. V-C, we use both 360-FoV images in spherical projection and fish-eye images in the unified camera model [28]. For the former, we collect our own dataset (named 360-MP3D-VO) by rendering the Matterport3D data [15] on Minos [16] to project a continues sequence of 360-FoV frames; for the latter, we use the TUM-VI dataset [17].

The evaluated baselines are the classic 8-PA [3] in spherical projection and the GSM solution [14]. For the former, we compare with our normalized approach using the optimization (14), which we refer to as $Opt\ \varepsilon(S,K)$. To compare with GSM, we use our proposed optimization (15), where we refer to ω_ξ and ω_{SK} as a weighting function evaluated by residuals from a camera pose ξ (computed by using the 8-PA[13]) and residuals obtained from our normalization $Opt\ \varepsilon(S,K)$, respectively.

Similar to [21], [11], [5], we use ε_R as the rotation error and ε_t as the translation error for camera pose estimation:

$$\varepsilon_R = \frac{1}{\pi}\, cos^{-1}\left(\frac{tr(\mathbf{R}^\top \tilde{\mathbf{R}}) - 1}{2}\right), \quad \varepsilon_\mathbf{t} = \frac{1}{\pi}\, cos^{-1}(\mathbf{t}^\top \tilde{\mathbf{t}})\ . \quad (16)$$

A. Noise and Outlier Evaluation

In this experiment, we project several landmarks in 3D by using the ground truth depth data from our 360-MP3D-VO dataset. To generate ground truth camera poses, we randomly sample translation vectors $\mathbf{t}$ in a uniform range of $[-1,1] \in \mathbb{R}^3$. Moreover, for each axis, we generate random rotation matrices $\mathbf{R} \in SO(3)$ by sampling Euler-angles in a uniform range of $[-\pi/4, \pi/4]$. Then, we construct our camera poses as homogeneous transformations $\mathbf{T} = [\mathbf{R}|\mathbf{t}] \in SE(3)$.

To show the effect of using different numbers of point, we uniformly sample between 8 to 200 3D-landmarks for each evaluation. Then, similar to [5], [11], we add a constant von Misses-Fisher noise (vMF) of $\kappa = 500$ (i.e., 3.21° of error) as well as a constant outlier ratio of 20%. The results of 15K evaluations are presented in Fig. 4-(a). In addition, to evaluate camera pose under different levels of noise, we sample 200 3D-landmarks of 15K different scenes, and then we incrementally add a vMF noise defined by a κ equals to 100, 200, 500, 1000, and 10000, representing 10.21°, 5.22°, 3.21°, 2.27°, 1.60° and 0.72° of error, respectively. This experiment is presented in Fig. 4-(b). To further evaluate our method against outliers, we use a constant number of 200 3D-landmarks and a vMF noise of $\kappa = 500$, then we increase outliers ratio from 0% to 70%. This is shown in Fig. 4-(c).

TABLE I

COMPARISONS WITH 8-PA METHODS.

	$\sigma_8\uparrow$	$\alpha\uparrow$	**Rot-e** $\times 10^{-3}$	**Tran-e** $\times 10^{-3}$	**Time (ms)**
$Opt\ \varepsilon(S,K)$	**20.025**	0.858	**11.429**	**25.416**	45.52
8-PA [3]	0.650	0.731	13.387	32.367	**40.33**
Isotropic (a) [13]	1.232	0.723	15.937	49.414	**40.33**
Non-Isotropic (b) [13]	0.012	0.723	13.439	34.354	**40.33**
(a) + (b) [22]	0.107	**0.879**	14.173	38.808	40.88

TABLE II

ABLATION STUDY.

		Rot-e $\times 10^{-3}$	**Tran-e** $\times 10^{-3}$	**Time (ms)**
GSM [14] (a)		9.1029	19.7902	45.318
Gaussian	(a) + not const ω_ξ	2.0951	5.2065	218.598
	(a) + not const ω_{SK}	**2.0943**	**5.1894**	212.924
	(a) + ω_ξ	3.9048	9.2592	**50.279**
	(a) + ω_{SK}	3.6802	7.5717	58.651
t-distribution	(a) + not const ω_ξ	9.1810	19.8160	**48.126**
	(a) + not const ω_{SK}	**8.9496**	**18.9284**	58.836
	(a) + ω_ξ	9.1810	19.8160	50.279
	(a) + ω_{SK}	9.0581	19.5573	53.795

In Fig. 4, we show favorable results against others in terms of noise level, outlier level, and number of points. For instance, our normalization is capable of reducing the effect of outliers around 10% compared with the baseline, i.e., using ω_ξ without our normalization to build ω.

B. Ablation Study

In this experiment, we show results of our solution by using tracked key-features from a sequence of 360-FoV images in a 6-DoF camera motion. For each frame in this sequence, we extract around 200 key-features using Shi-Tomasi key-points [29] and the KLT tracker [30]. For every evaluation, we ensure that there exits at least a minimum distance of 0.5m between frames. We evaluate over 2K different environments and compute the median values of the estimated errors.

In Table I, we compare the proposed normalized solution with the 8-PA algorithms [3], [13], [22], all of them in spherical projection. Note that isotropic and non-isotropic pre-conditioning are the normalization strategies proposed by [13], which are particularly used for uncalibrated cameras in perspective view, but can still be used for spherical projection. In the results, our normalization is capable of reducing errors in camera pose without significantly adding the overhead time. Note that, our normalization obtains larger values of σ_8 (second least singular value of an observation matrix $\mathbf{A}$) as well as α (motion parallax in the normalizing domain), showing that our normalization truly increases the

TABLE III

EXPERIMENTAL RESULTS IN REAL SCENES ON MP3D-VO AND TUM-VI.

		Rotation error $\times 10^{-3}$					**Translation error** $\times 10^{-3}$				
		8-PA [3]	*Opt* $\varepsilon(S,K)$	GSM [14]	ω_ξ	ω_{SK}	8-PA [3]	*Opt* $\varepsilon(S,K)$	GSM [14]	ω_ξ	ω_{SK}
MP3D-VO	Q75	23.577	**20.504**	15.286	9.909	**7.964**	91.442	**72.693**	43.707	29.945	**20.653**
	Q50	10.827	**9.515**	7.389	3.523	**3.072**	36.814	**29.645**	19.453	9.985	**7.966**
	Q25	4.624	**4.075**	3.277	1.638	**1.524**	14.745	**12.276**	8.501	4.611	**4.009**
TUM-VI [17]	Q75	44.479	**40.514**	49.291	39.938	**38.256**	221.131	**199.042**	166.594	169.122	**140.846**
	Q50	27.211	**24.969**	27.226	21.075	**19.241**	117.701	**104.742**	82.457	74.661	**62.743**
	Q25	15.512	**14.462**	14.595	10.507	**9.708**	58.791	**53.283**	40.899	33.724	**28.800**

TABLE IV

EXPERIMENTAL RESULTS UNDER DIFFERENT THRESHOLDS WITH RANSAC.

	Threshold: 2.30E-04 **Iterations:** 590 **Outliers in data:** 1%					**Threshold:** 1.10E-03 **Iterations:** 66 **Outliers in data:** 20%				
	8-PA [3]	*Opt* $\varepsilon(S,K)$	GSM [14]	ω_ξ	ω_{SK}	8-PA [3]	*Opt* $\varepsilon(S,K)$	GSM [14]	ω_ξ	ω_{SK}
Rot-e $\times 10^{-3}$	3.957	**3.950**	3.691	2.994	**2.919**	10.658	**9.963**	9.978	4.292	**4.065**
Tran-e $\times 10^{-3}$	8.757	**8.689**	8.239	7.660	**7.477**	22.656	**19.806**	20.394	10.540	**9.452**
Residual	**2.72E-06**	**2.72E-06**	**2.88E-06**	2.94E-06	2.94E-06	**2.72E-05**	2.74E-05	**2.77E-05**	2.91E-05	2.92E-05
Time (sec)	**0.149**	0.153	**0.181**	0.183	0.189	**0.015**	0.021	**0.050**	0.065	0.055

stability in the DLT solution of an essential matrix.

In Table II, we compare the effect of our normalization in the weighted non-linear optimization (15) upon the GSM solution [14]. In rows 2-5, we find that using a Gaussian distribution constantly performs better than t-distributions. In rows 2,3 and 6,7, we evaluate a non-constant weighing function ω which is updated at every iteration in the LM optimization [31], [32], [5]; this approach is also known as Iterative Re-weighted Least-Square method (IRLS). Although IRLS achieves the lowest error, it is still the most time-consuming method among all evaluations.

C. Experiments on Real Scenes

We evaluate a sequence of 360-FoV and fish-eye images with our own 360-MP3D-VO and the TUM-VI [17] datasets, respectively. Similar to Sec. V-B, we track 200 key-features between frames, evaluating around 15K pairs of frames using 360-MP3D-VO, and 16K paired images using TUM-VI.

For the evaluations on the TUM-VI [17] dataset, we only use the scenes that contain a complete ground truth camera pose, i.e., room-1 to room-6. Moreover, since this dataset is mainly used for visual-inertial tasks, some frames in our evaluation has been skipped due to drastic camera movements and severe changes in illumination.

Results of camera pose errors on both datasets are presented in Table III, using the quantiles Q25, Q50 and Q75 of error evaluations. From the results, we can verify that our approaches constantly outperform the baselines, with the lowest errors in every evaluation. Moreover, for the averaged condition (i.e., Q50 results), our strategies are capable of reducing errors up to 10% by normalizing bearing vectors, using *Opt* $\varepsilon(S,K)$; and up to 50% by using our weighted optimization ω_{SK}.

Additionally, we evaluate our proposed approaches *Opt* $\varepsilon(S,K)$ and ω_{SK} in the context of a RANSAC evaluation. We design two settings using the same amount of correspondence bearing vectors, noise, and outliers (e.g., 400 3D-landmark, vMF noise of $\kappa = 500$, and 50% of outliers), but with two different thresholds for RANSAC. In addition, upon the RANSAC results, we evaluate the final essential matrix using only the detected inliers by using our proposed methods as well as the baselines. In Table IV, results over 2K evaluations are presented.

On the left side of Table IV, we set a threshold for RANSAC as 2.3×10^{-4}, which successfully detects all the inliers after 590 iterations; whereas on the right side, the previous threshold is relaxed until 1.1×10^{-3}, speeding up the evaluation by detecting 70% of inliers in 66 RANSAC iterations. However, since we know in advance the outlier ratio of our data, we can assert that there is 20% of the remaining outliers. Therefore, comparing columns 2 and 11, we verify that our proposed ω_{SK} performs similar to the one using RANSAC with a tight threshold, but 3 times faster, showing the benefit of our robust estimation.

VI. CONCLUSIONS

In this paper, we propose a novel pre-conditioning strategy to the classic 8-point algorithm [3] for estimating an essential matrix in spherical projection. Our solution redesigns the normalizing algorithm [13] to alleviate the effect of poor/uneven distribution of key-features, increasing stability and robustness against outliers. We also extend our approach, towards improving the well-known Gold Standard Method [14] for spherical projection by proposing a constant weighted non-linear optimization, built upon our normalization strategy. Based on extensive experiments under different scene conditions, our proposed methods outperform the baselines, increasing the accuracy in camera pose up to 30% without a significant impact on the computation time.

Acknowledgement. This project is supported by MOST Joint Research Center for AI Technology and All Vista Healthcare with program MOST 110-2634-F-007-016 and MOST 110-2636-E-009-001.

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Tutorial: visual odometry," *IEEE Robotics and Automation Magazine (RA-M)*, 2011.

[2] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[3] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, 1981.

[4] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2004.

[5] H. Guan and W. A. P. Smith, "Structure-from-motion in spherical video using the von mises-fisher distribution," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 711–723, 2017.

[6] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*, 2016.

[7] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework," in *ACM Conference on Multimedia (MM)*, 2019.

[8] C. Fermüller and Y. Aloimonos, "Ambiguity in structure from motion: Sphere versus plane," *International Journal of Computer Vision (IJCV)*, 1998.

[9] J. Fujiki, A. Torii, and S. Akaho, "Epipolar geometry via rectification of spherical images," in *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, 2007.

[10] H. Taira, Y. Inoue, A. Torii, and M. Okutomi, "Robust feature matching for distorted projection by spherical cameras," *IPSJ Transactions on Computer Vision and Applications*, 2015.

[11] T. L. da Silveira and C. R. Jung, "Perturbation analysis of the 8-point algorithm: a case study for wide fov cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 757–11 766.

[12] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine (RA-M)*, 2012.

[13] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1997.

[14] A. Pagani and D. Stricker, "Structure from motion using full spherical panoramic cameras," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 375–382.

[15] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," in *International Conference on 3D Vision (3DV)*, 2017.

[16] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "MINOS: Multimodal indoor simulator for navigation in complex environments," *ArXiv:1712.03931*, 2017.

[17] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *International Conference on Intelligent Robots and Systems (IROS)*, October 2018.

[18] Hongdong Li and R. Hartley, "Five-point motion estimation made easy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1, 2006, pp. 630–633.

[19] H. Li and R. Hartley, "Five-point motion estimation made easy," in *International Conference on Pattern Recognition (ICPR)*, 2006.

[20] B. Li, L. Heng, G. H. Lee, and M. Pollefeys, "A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[21] K. Fathian, J. P. Ramirez-paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "QuEst: A Quaternion-Based Approach for Camera," *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[22] M. Mühlich and R. Mester, "The role of total least squares in motion analysis," in *European Conference on Computer Vision*. Springer, 1998, pp. 305–321.

[23] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer Science & Business Media, 2012, vol. 26.

[24] A. Torii, A. Imiya, and N. Ohnishi, "Two-and three-view geometry for spherical cameras," in *Proceedings of the sixth workshop on omnidirectional vision, camera networks and non-classical cameras*, 2005.

[25] P.-Å. Wedin, "Perturbation bounds in connection with singular value decomposition," *BIT Numerical Mathematics*, vol. 12, no. 1, pp. 99–111, 1972.

[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[27] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004, [Accessed on 31 Jan. 2005.].

[28] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 552–560.

[29] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features track better," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*. IEEE, 1998, pp. 178–183.

[30] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," *International Joint Conferences on Artificial Intelligence*, 1981.

[31] P. H. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International journal of computer vision*, vol. 24, no. 3, pp. 271–300, 1997.

[32] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3748–3754.