2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Model Quality Aware RANSAC: A Robust Camera Motion Estimator

Shu-Hao Yeh, Yan Lu, and Dezhen Song

***Abstract*— Robust estimation of camera motion under the presence of outlier noise is a fundamental problem in robotics and computer vision. Despite existing efforts that focus on detecting motion and scene degeneracies, the best existing approach that builds on Random Consensus Sampling (RANSAC) still has non-negligible failure rate. Since a single failure can lead to the failure of the entire visual simultaneous localization and mapping, it is important to further improve the robust estimation algorithm. We propose a new robust camera motion estimator (RCME) by incorporating two main changes: a model-sample consistency test at the model instantiation step and an inlier set quality test that verifies model-inlier consistency using differential entropy. We have implemented our RCME algorithm and tested it under many public datasets. The results have shown a consistent reduction in failure rate when comparing to the RANSAC-based Gold Standard approach and two recent variations of RANSAC methods.**

I. Introduction

Robust estimation of geometric relationships between two camera views is a fundamental problem in computer vision and robotics. It simultaneously identifies corresponding inlier features from outlier noises. When applied to the problem of visual odometry (VO) or visual simultaneous localization and mapping (vSLAM) in robotics, the geometric relationship is often the fundamental matrix, or essential matrix when camera intrinsic parameters are known. Since camera motion can be inferred from the essential matrix, this is also known as camera motion estimation.

The classical robust estimation method that can filter out outliers is random consensus sampling (RANSAC) [13] (Fig. 1(a)). Simply applying RANSAC to estimate camera motion is not wise since there are well-known motion degeneracy and scene degeneracy issues, also known as Type A and Type B degeneracies according to [39], respectively. To eliminate both degeneracies, existing state-of-the-art approaches apply key frame selection and simultaneously estimate homography and fundamental matrices [25]. However, these remedies do not solve every problem. After testing 46643 images from 23 different datasets with 3 iteration number settings, our results show that the RANSAC-based fundamental matrix estimation algorithm still has a failure rate of 1.83% in indoor environments at iteration number of 1000, despite test data having both degeneracies removed. Since one single failure can lead to an entire continuous monocular vSLAM failure, the mean distance between failure is mere 49 meters if the average distance between adjacent key frames is 0.5 meters. This means that current methods for the estimation of camera motion is not robust enough.

*This work was supported in part by NSF under NRI- 1925037.

S. Yeh and D. Song are with CSE Department, Texas A&M University, College Station, TX 77843, USA. Emails: *ericex1015@tamu.edu* and *dzsong@cs.tamu.edu*.

Y. Lu is with Google AR/VR team, 1600 Amphitheatre Parkway Mountain View, CA 94043, USA. Email:*sinoluyan@gmail.com*.

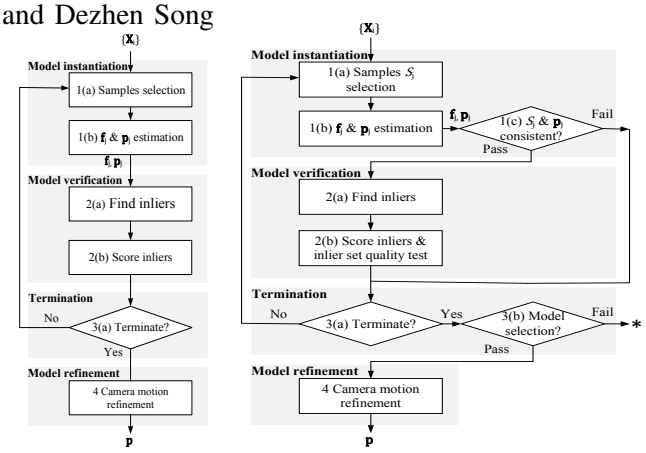

Fig. 1. The system diagrams of our RCME framework in (b) which is an improvement of RANSAC in (a), the Gold standard algorithm for estimating camera motion.

A deep look into the failed cases reveals that since the epipolar geometric relationship characterized by a fundamental matrix is a weak point-to-line distance measure and scene feature distribution may not be uniform, it is possible that the initial sampling in RANSAC may establish a wrong model and still can find a potential large number of inliers. Hence RANSAC may output incorrect camera motion. To address these issues, we propose a new robust camera motion estimator (RCME) (see Fig. 1(b)) by incorporating two main changes: model-sample consistency test to verify the quality of the model built by over-parameterized sampling, and inlier set quality test that first determines inliers according to model uncertainty and then verifies if the inlier set agrees with the model through a test on differential entropy. The RCME can also detect and report a failed estimation when input data quality is too low. We have implemented the RCME algorithm in C++ and tested with public datasets. The results have shown consistent reduction of failure rate across almost all test data. It also have been compared to two recent RANSAC variations and outperforms the both counterparts.

II. Related Work

Robust camera motion estimation mainly relates to two research fields: robust estimation in computer vision and the front-end algorithm of vSLAM/VO in robotics.

Robust estimation is a fundamental problem in computer vision and robotics. A robust estimator's task is to estimate parameters and find the inliers according to a predefined type of geometric relationship. Certifying the solution also can enhance the estimation [42]. The outliers from wrong correspondences often introduce significant errors when estimating the geometric relationship. M-estimator, L-estimator and R-estimator [19] formulate the estimation problem as a reprojection error minimization problem and solve it by using a nonlinear cost function. Leat Median of Squares (LMeds)

by Rousseeuw [32] minimizes the median of error instead. However, these methods are not stable when over half of data are outliers since they perform the minimization among all the data including the outliers. RANSAC by Fischler et al. [13] is the most widely used robust estimator since it is capable of handling a high ratio of outliers.

The vSLAM and VO works can be classified as the feature-based approaches [9], [21], [25], the semi-dense approaches [12], [14], and the direct approaches [10], [11]. Among these methods, the widely adopted feature-based approaches explicitly and repeatedly utilize robust estimation of camera motion (e.g., MonoSLAM [9], PTAM [21] and ORB-SLAM2 [25]). RANSAC has been employed in combination with different fundamental matrix estimation models such as the one-point method in [34], the five-point algorithm [35] in PTAM or the eight-point method in ORB-SLAM2. These works repeatedly obtain initial solutions for camera poses (key) frame by (key) frame before bundle adjustment. Hence the robustness of the robust estimation method is critical here because a single failure can collapse the entire mapping.

Our work is an improvement over the existing RANSAC framework for camera motion estimation. In fact, most existing works in this area mainly focus on detecting the two type of degeneracy issues: Type A and Type B [39]. When handling degeneracy cases, existing approaches often treat the problem as a model selection between the homography relationship and the fundamental matrix relationship. Popular approaches include AIC [1], PLUNDER-DL score [41], GIC [20], and GRIC [38]. Pollefeys et al. [28] uses GRIC in recovering the structure and motion. DEGENSAC [8] sets up the H-degeneracy test for the degeneracy issues, and incorporates the homography in the epipolar geometry estimation if the degeneracy is detected. ORB-SLAM2 initializes the trajectory by finding the better geometry model between the fundamental matrix or the homography matrix. However, even when Type A and Type B degeneracies have been ruled out, our tests have shown that RANSAC-based camera motion estimation may still fail with non-negligible probability. It is due to the fact that the fundamental matrix only represents a weak geometric point-line distance geometric relationship which is not very selective.

Many attempts have been tried to improve RANSAC in general. Guided sampling replaces the random sampling by employing the prior information [6], [29], [36] or the heuristics [26], [31], [37]. The statistic testing including the $T_{d,d}$ test [5], the Bail-out test [4] and the SPRT test [24], [29] to evaluate the hypothesis provides an early evaluation. LO-RANSAC [7] adds the inner RANSAC in the model refinement to find better support of inliers. QDEGSAC [15] can handle the (quasi-)degenerate data by involving in the robust rank detection. However, these methods are not stable in camera motion recovery since they are designed for general geometric model instead of the fundamental matrix.

Our work emphasizes the robustness in camera pose estimation. We incorporate the model uncertainty in every step of RANSAC framework. We employ the information entropy for the hypothesis evaluation to improve model quality and hence increase robustness. The threshold used in determining inliers is based on the model uncertainty. MLESAC [40] introduces the loss function in hypothesis selection by modeling inliers and outliers respectively. It has been embedded in many state-of-the-arts (e.g., the initialization module of ORB-SLAM2). Cov-RANSAC [30] which also incorporates the model uncertainty to identify inliers, but they do not rank the model by using the model-inlier consistency.

III. Problem Definition

We begin with the following assumptions:

a.1 The camera lens distortion removed.

a.2 Position noises of the points follow zero-mean Gaussian distribution with known variance σ^2 in each dimension and the noise in each dimension is independent.

Let us define the common notations in this paper.

- K the intrinsic matrix of the camera.
- $\mathbf{X}$ is a 4-vector and concatenates point correspondences from two views. The i-th point is denoted as $\mathbf{X}_i = \begin{bmatrix}\tilde{\mathbf{x}}_i^\mathsf{T} & \tilde{\mathbf{x}}_i'^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^4$, where $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_i'$ is the i-th corresponding points from the first and the second views, respectively. Symbol $\tilde{}$ indicates the inhomogeneous coordinate and $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_i' \in \mathbb{R}^2$. The corresponding 3-vector homogeneous representation is $\mathbf{x}_i, \mathbf{x}_i' \in \mathbb{P}^2$.
- $\mathbf{f}$ is a 9-vector with entries from fundamental matrix F, $\mathbf{f} = \begin{bmatrix}\mathrm{F}^1 & \mathrm{F}^2 & \mathrm{F}^3\end{bmatrix}^\mathsf{T} \in \mathbb{R}^9$, where F^i denotes the i-th row of F. The i-th model vector is $\mathbf{f}_i$.
- $\mathbf{p}$ is a 7-vector in $\mathrm{SE}(3)$ defining the camera motion and consists of a unit quaternion vector $\mathbf{q} \in \mathbb{R}^4$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. The i-th camera motion denotes as $\mathbf{p_i} = \begin{bmatrix}\mathbf{q}_i^\mathsf{T} & \mathbf{t}_i^\mathsf{T}\end{bmatrix}^\mathsf{T}$.

In this paper, I_n denotes the $n \times n$ identity matrix.

We pre-process both views to obtain inputs. Feature detection and feature matching have been applied to obtain putative point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}_{i=1}^n$, where n is the number of point correspondences. We concatenate every inhomogeneous point correspondence $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_i'$ in to a 4-vector $\mathbf{X}_i = \begin{bmatrix}\tilde{\mathbf{x}}_i^\mathsf{T} & \tilde{\mathbf{x}}_i'^\mathsf{T}\end{bmatrix}^\mathsf{T}$. The inputs may be obtained from a variety of feature detectors such as scale-invariant feature transform (SIFT) [22], speeded up robust feature (SURF) [2], or ORB [33]. It is worth noting that the point correspondence set is the result of putative matching of feature descriptors and often contains many outliers. Our problem is,

Definition 1: Given n point correspondences (or points for brevity) $\{\mathbf{X}_i\}_{i=1}^n$, determine if the solution of the camera motion $\mathbf{p}$ exists. If the solution exists, estimate $\mathbf{p}$.

As shown in the problem definition, one immediate difference between our method and RANSAC is that our method can detect failure cases instead of output an unreliable solution. This is actually very useful for vSLAM because it can be used as a signal to re-adjust key frame selection.

IV. RCME Aglorithm

A. RANSAC Review and RCME Overview

Since our RCME is an extension of RANSAC, let us begin with a brief review of RANSAC as shown in Fig. 1(a).

RANSAC establishes hypothesis models by randomly sampling a minimal correspondence set (model instantiation) and then examines each set by comparing how many other features agree with the hypothesis model (model verification). RANSAC is an iterative method which terminates when reaching a maximum iteration number or a high quality inlier set with its size agreeable to the estimated inlier ratio is found. The model with the largest inlier set is the output.

Building on basic RANSAC framework, Fig. 1(b) illustrates our system diagram. It contains four main blocks in gray. The first three blocks are different from the counterpart. After instantiating a hypothesis model through sampling, we test the consistency between the samples and the model. If no consistency exists between the samples and the model, we discard the model and repeat the model instantiation step. Otherwise, we begin model verification, where we perform an inlier quality test before we select the model according to entropy. If no model can be selected, we consider the camera motion estimation as failure in termination step (See $*$ from Box 3(b) in Fig. 1(b)). The selected model is our initial solution for the model refinement.

Without loss of generality, let us assume we are at the j-th iteration to begin the explanation.

B. Model Instantiation

Model instantiation is the first step of every iteration. We want to check if a model building on the randomly sampled point correspondences actually agrees with the samples. This is crucial for overpameterized models.

Same as the traditional RANSAC, we randomly sample m point-correspondences from $\{\mathbf{X}_i\}$ where m is the minimal number of samples to instantiate the model and can be different according to different parameterizations used in modeling. Define the sample set for the j-th iteration as $\mathcal{S}_j := \left\{\mathbf{X}_{s_k} : s_k \in \{1, \cdots, n\} \quad \text{and} \quad 1 \leq k \leq m\right\}$, where s_k is the point index and k denotes the k-th sample.

1) $\mathbf{f}_j$&$\mathbf{p}_j$ model instantiation and uncertainty analysis: Given the sample set $\mathcal{S}_j$, we instantiate the model $\mathbf{f}_j$ and recover the camera motion $\mathbf{p}_j$ from $\mathbf{f}_j$. The camera motion $\mathbf{p}_j = \begin{bmatrix}\mathbf{t}_j^\mathsf{T} & \mathbf{q}_j^\mathsf{T}\end{bmatrix}^\mathsf{T}$ and the fundamental matrix model $\mathbf{f}_j = \begin{bmatrix}\mathrm{F}_j^1 & \mathrm{F}_j^2 & \mathrm{F}_j^3\end{bmatrix}^\mathsf{T}$ always satisfy $[\mathbf{t}_j]_\times R(\mathbf{q}_j) = s\mathrm{K}^\mathsf{T}\begin{bmatrix}\mathrm{F}_j^1 \\ \mathrm{F}_j^2 \\ \mathrm{F}_j^3\end{bmatrix}\mathrm{K}$, where s is a scalar, $[\mathbf{t}_j]_\times$ is the skew-symmetric matrix representation of $\mathbf{t}_j$, and $R(\mathbf{q}_j)$ is the rotation matrix of the unit quaternion vector $\mathbf{q}_j$. The camera motion $\mathbf{p}_j$ is recovered by using the fundamental matrix decomposition [18].

Consider a mapping function Ω which maps the camera motion $\mathbf{p}_j$ to the concatenated samples $\left[\ldots, \mathbf{X}_{s_k}^\mathsf{T}, \ldots\right]^\mathsf{T}$, $\Omega : \mathbb{R}^7 \rightarrow \mathbb{R}^{4m}$. The concrete representation of Ω depends on parameterization of fundamental matrix which is utilized in the camera motion recovery. It includes $1-$point [34], $5-$point [27], or $8-$point methods [18]. Since our framework is not limited by a particular parameterization, we represent it as a generic Ω mapping. Therefore, the camera motion is

$$\mathbf{p}_j = \Omega^{-1}([\ldots, \mathbf{X}_{s_k}^\mathsf{T}, \ldots]^\mathsf{T}). \tag{1}$$

Uncertainty analysis. The uncertainty of $\mathbf{p}_j$ depends on the error distribution of model instantiation samples $\left[\ldots, \mathbf{X}_{s_k}^\mathsf{T}, \ldots\right]^\mathsf{T}$ and is propagated through the mapping function $\Omega^{-1}(\cdot)$. The noise distribution of $\mathbf{X}_{s_k}$ is modeled as a zero-mean Gaussian with the covariance matrix $\Sigma_X = \sigma^2\mathrm{I}_4$ according to the assumption **a.2**.

We utilize the first order approximation [18] of the covariance matrix to estimate the uncertainty. Denote the covariance matrix of $\mathbf{p}_j$ as Σ_{p_j}. The first order approximation of Σ_{p_j} is $\Sigma_{p_j} = \left(J_\Omega^\mathsf{T}\Sigma_S^{-1}J_\Omega\right)^{-1}$, where $\Sigma_S = diag(\cdots, \Sigma_X, \cdots)$ and Jacobian matrix $J_\Omega = \frac{\partial\Omega}{\partial\mathbf{p}_j}$.

2) Samples and $\mathbf{p}_j$ consistency test: A good model must be consistent with the samples that instantiate it. However, the consistency between the samples and the instantiated model can be always satisfied if the model is parameterized by using the same amount of parameters as the degrees of freedom (DoFs) of the model. Before assessing the consistency, it is necessary to know how fundamental matrix estimation is parameterized:

> **Exact case**: The fundamental matrix is parameterized by the same amount of parameters as the DoFs of fundamental matrix. For example, the 1-point algorithm is an exact case since the fundamental matrix is parameterized by one yaw angle and the DoF of fundamental matrix is 1 when the robot motion is assumed to follow the Ackermann steering model on a planar surface.
>
> **Over-parameterized case**: The amount of parameters used to parameterize the fundamental matrix is larger than the DoFs of fundamental matrix. For example, the normalized 8-point algorithm is a common Ooer-parameterized case since it employs 8 parameters but the DoFs of a general fundamental matrix is 7.

For the over-parameterized case, we perform the consistency test between the samples and the model. Only the model which passes this consistency test can advance to the next step. Otherwise, we discard the model and the j-th iteration ends. Of course, this does not apply to the exact case where its model always perfectly fits the samples.

The consistency between the samples and the model is measured by the error distance. We use the Sampson error vector [18] to form the error measurement. Denote the Sampson error vector of $\mathbf{X}_{s_k}$ as $\boldsymbol{\delta}_{s_k}$. Let Δ as the Sampson error vector function which utilizes $\mathbf{p}_j$ to calculate the Sampson correction of $\mathbf{X}_{s_k}$ as $\boldsymbol{\delta}_{s_k} = \Delta(\mathbf{X}_{s_k}, \mathbf{p}_j) \in \mathbb{R}^4$. We model $\boldsymbol{\delta}_{s_k}$ as the zero-mean Gaussian distribution with the covariance $\Sigma_{\delta_{s_k}}$. Under the Gaussian noise assumption, the first-order approximation $\Sigma_{\delta_{s_k}} = J_{\delta,X}\Sigma_X J_{\delta,X}^\mathsf{T} + J_{\delta,p}\Sigma_{p_j}J_{\delta,p}^\mathsf{T}$, where Jacobian matrices $J_{\delta,X} = \frac{\partial\Delta}{\partial\mathbf{X}_{s_k}}$ and $J_{\delta,p} = \frac{\partial\Delta}{\partial\mathbf{p}_j}$.

For each sample $\mathbf{X}_{s_k}$, we design hypothesis testing:

$$\mathbf{H_0} : \mathbf{X}_{s_k} \text{ does not fit} \mathbf{p}_j, \qquad \mathbf{H_1} : \text{Otherwise.} \tag{2}$$

Given the Sampson error vector $\boldsymbol{\delta}_{s_k}$ and the covariance matrix $\Sigma_{\delta_{s_k}}$, the error distance is re-written as

$$\boldsymbol{\delta}_{s_k}^\mathsf{T}\Sigma_{\delta_{s_k}}^{-1}\boldsymbol{\delta}_{s_k}. \tag{3}$$

Since we approximate $\boldsymbol{\delta}_{s_k}$ as the normal distribution with zero mean vector and the covariance $\Sigma_{\delta_{s_k}}$, (3) is a χ^2 distribution. Besides, $\boldsymbol{\delta}_{s_k}$ is defined on the variety of $\mathbf{x}_{s_k}^{\mathsf{T}}\mathrm{K}^{-\mathsf{T}}[\mathbf{t}_j]_\times R(\mathbf{q}_j)\mathrm{K}^{-1}\mathbf{x}_{s_k}$, which reduces 1 DoF. Therefore, (3) follows a χ^2 distribution with $4-1=3$ DoFs. Define G_3 as the cumulative χ^2 distribution under 3 DoFs and we can set the distance threshold $G_3^{-1}(1-\alpha)$ by setting the significance level $\alpha = 0.05$, where $G_3^{-1}(\cdot)$ is the inverse function of $G_3(\cdot)$. Thus, we consider $\mathbf{X}_{s_k}$ agrees with $\mathbf{p}_j$ by rejecting $\mathbf{H_0}$ when $\boldsymbol{\delta}_{s_k}^{\mathsf{T}}\Sigma_{\delta_{s_k}}^{-1}\boldsymbol{\delta}_{s_k} \leq G_3^{-1}(1-\alpha)$.

C. Model Verification

For the model which passes the aforementioned consistency tests, we find inliers from the rest of the inputs and verify the model by checking the quality of its inliers (See Boxes 2(a) and 2(b) in Fig. 1(b)).

1) Find inliers: An inlier is defined to be a point consistent with the model. We employ $\mathbf{p}_j$ consistency test to find the inliers.

For the point $\mathbf{X}_i$, the Sampson error vector and corresponding covariance are denoted as $\boldsymbol{\delta}_i$ and Σ_{δ_i}, respectively, and can be obtained from their definition. Let I_j as an inlier indicator function of $\mathbf{p}_j$

$$I_j(\mathbf{X}_i) := \begin{cases} 1, & \text{when} \quad \boldsymbol{\delta}_i^{\mathsf{T}}\Sigma_{\delta_i}^{-1}\boldsymbol{\delta}_i \leq G_3^{-1}(1-\alpha) \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The inlier set of $\mathbf{p}_j$ is $\mathcal{X}_j := \left\{\mathbf{X}_{i_k} : I_j(\mathbf{X}_{i_k}) = 1, \quad 1 \leq i_k \leq n\right\}$ with its size defined as $n_j = |\mathcal{X}_j|$.

2) Score inliers: It measures the quality of the consistency between model and its inlier set. Instead of using n_j, the number of inliers or the loss function [40] to score the inliers, we want to score inliers by using the differential entropy on the covariance matrices $\Sigma_{\delta_{i_k}}$. The intuition is that the joint distribution of distances $\{\boldsymbol{\delta}_{i_k}, k = 1, 2, ..., n_j\}$ for the inliers should have a small entropy for a high quality inlier set. For inlier set $\mathcal{X}_j$, we define a score vector $\mathbf{h}_j$ as,

$$\mathbf{h}_j = [\ldots, h_{i_k,j}, \ldots]^{\mathsf{T}} \in \mathbb{R}^{n_j}, \tag{5}$$

where each entry is a differential entropy for each inlier, $h_{i_k,j} = \frac{1}{2}\log((2\pi)^4 \exp(4)|\Sigma_{\delta_{i_k}}|)$ where $|\Sigma_{\delta_{i_k}}|$ denotes the determinant of $\Sigma_{\delta_{i_k}}$.

3) Inlier set quality test: Now, we evaluate the inliers' quality to determine if the model can enter candidate solution set by checking entropy values. A good model must contain the inliers with high quality of consistency and leads to small entropy values. Given the score vector $\mathbf{h}_j$, the average entropy and the standard deviation are defined as follows: $\psi_j = \frac{\|\mathbf{h}_j\|_1}{n_j}$, $s_j = \sqrt{\frac{\|\mathbf{h}_j - \psi_j \mathbf{1}_n\|_2^2}{n_j - 1}}$, where $\|\cdot\|_1$ is L1 norm, $\|\cdot\|_2$ is L2 norm, and $\mathbf{1}_n$ is a n-vector of ones.

To evaluate the quality of the model and its inlier set, we design the following hypothesis testing based on the $Z-$test,

$$\mathbf{H_0} : \psi_j > \mu, \qquad \mathbf{H_1} : \text{Otherwise}, \tag{6}$$

where $\mu = 2.90$ is an differential entropy threshold determined by the experiments. The test statistic can be calculated $Z_j = \frac{\psi_j - \mu}{s_j/\sqrt{n_j}}$. Define $\Phi(x)$ as the cumulative distribution function of the standard normal distribution at value x. By setting the significance level α, the p-value is obtained $\Phi^{-1}(1-\alpha)$. We consider that the model is highly consistent with its inlier set by rejecting $\mathbf{H_0}$ when $Z_j \leq \Phi^{-1}(1-\alpha)$. For models that passed the hypothesis testing, we proceed to next step.

In addition to the high quality inliers, a good model also need to contain sufficiently large amount of inliers. We use two ratio thresholds ω_p and λ to determine if the j-th model satisfies the requirement. ω_p is a priorly-known or estimated inlier ratio. A model is considered to contain a sufficiently large amount of inliers when $\frac{n_j}{n} \geq \lambda\omega_p$, where conservative coefficient $0.5 \leq \lambda \leq 1$ determine how close to ideal size ω_p we want the inlier set to be. Same as the RANSAC, inlier ratio ω_p can be inferred in the process. All models surviving the hypothesis testing in (6) and being sufficiently large are added to the candidate solution set,

$$\mathcal{C} := \left\{j | \left(Z_j \leq \Phi^{-1}(1-\alpha)\right) \ \wedge \ \left(\frac{n_j}{n} \geq \lambda\omega_p\right), \forall j\right\}. \tag{7}$$

After the maximum number of trials N, we consider the camera motion recovery fail if $\mathcal{C} = \emptyset$. This leads to the failed algorithm output (i.e. '$*$' in Fig. 1(b)). Provided sufficiently large iteration number, the failure reason is most due to poor quality inputs (PQI) in $\{\mathbf{X}_i\}$.

D. Model Selection and Early Termination

1) Model selection: For non empty $\mathcal{C}$, we select the one with the minimum average entropy to be the output.

$$\min_{j \in \mathcal{C}} \frac{\|\mathbf{h}_j\|_1}{n_j}. \tag{8}$$

This output serves as an initial solution for the following model refinement by applying Maximum Likelihood Estimation (MLE) to minimize reprojection error. Since this is the same as the traditional approach (i.e. Box 4s in both Fig. 1(a) and Fig. 1(b), we skip it here.

2) Early termination condition: So far, our RCME algorithm runs for the entire maximum iterations. It is possible to design an early termination threshold to speed up the algorithm. Note that entries in $\mathcal{C}$ grow after each successful iteration. For each new entry, we can test its average entropy $\frac{\|\mathbf{h}_j\|_1}{n_j}$ by comparing to a preset threshold. If it is small enough, which means it is a satisfying solution, we can terminate RCME early to perform model refinement.

V. EXPERIMENTS

We have implemented our system in C++. We first evaluate the camera pose accuracy using the simulated experiments and commonly-used public data sets.

A. Algorithms in Comparison and Settings

We compare our algorithm to the following approaches.

- Standard: the most widely-used Gold standard approach with RANSAC [13], [18].
- MLESAC [40]: a representative RANSAC-based method in using the loss function to score inliers.

- Cov-RANSAC [30]: a representative RANSAC-based method employing the model uncertainty to verify the inliers. The model with the most inliers' support wins.

We also include "pRCME" which is RCME with model consistency test (2) in Sec. IV-B.2 turned off. The purpose is to show the individual effectiveness of the consistency test and the following inlier set quality test.

All algorithms employ the normalized 8-point algorithm [18] for the fundamental matrix estimation since it is fast and works for general camera motion. The maximum iteration number N is set according to different experiments and we perform a complete N iterations instead of estimating N during the iteration process. Best results are highlighted in boldface fonts in the tables.

B. Simulated Experiments

In this simulation two cameras are in a corridor surrounding by 5 walls. Each camera is a monocular camera, where the camera focal length is 500 and the image resolution is 640×480. Both cameras are free-flying cameras. We fix the baseline distance by 1 meter and set the depth of the corridor with 50 meters to avoid both Type A and B degeneracy [39]. Before we obtain a pair of two-view images, we decide the number of points n, inlier ratio ω and the injected noise σ on the images. Then we uniformly sample the points on the walls and project them on images with the noise and the outliers.

Given the inlier ratio ω, the maximum iteration number N is determined by $N = \log(1-p)/\log(1-\omega^s)$, where p is the probability that ensure at least one of the s samples is free from outliers. p is usually chosen at 0.99 and $s = 8$ since we choose the normalized 8-point algorithm [18] for the fundamental matrix estimation.

1) Evaluation metric for accuracy: To capture the accuracy of the camera motion, we evaluate the the orientation and the translation separately. Given the estimate camera motion $\mathbf{p} = \begin{bmatrix}\mathbf{q}^\mathsf{T} & \mathbf{t}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ and the ground truth $\bar{\mathbf{p}} = \begin{bmatrix}\bar{\mathbf{q}}^\mathsf{T} & \bar{\mathbf{t}}^\mathsf{T}\end{bmatrix}^\mathsf{T}$, where $\mathbf{q}$ is unit quaternion vector and the unit of $\mathbf{t}$ is meter, the orientation error Δq [17] and the translation error Δt are defined as $\Delta q := \min\left(\|\mathbf{q} - \bar{\mathbf{q}}\|_2, \|\mathbf{q} + \bar{\mathbf{q}}\|_2\right)$, and $\Delta t := \|\mathbf{t} - \bar{\mathbf{t}}\|_2$, respectively.

2) Experimental results: Table I shows the average and standard deviation of 5000 two-view pairs of images. Both RCME and pRCME outperform the "Standard" approach, MLESAC and Cov-RANSAC especially in the rotation error. RCME and pRCME are about 10 times smaller in the rotation error compared with the other three approaches. This is because that RCME and pRCME employ the average entropy to measure the quality of the inlier set instead of the amount of inliers in the model selection. The "Standard" approach and Cov-RANSAC utilizes the inlier amount to score the hypothesis. Counting the inlier amount is not reliable when the poor configuration of the data happens. For example, the indoor environment dominated by the planar surfaces like the simulated corridor in the experiments often contains highly degeneracy configuration even though we already eliminate the Type A and B degeneracy from our data. The highly degenerate configuration leads to the fact that false hypotheses can gain as much as support compared with the true hypothesis, and eventually the false hypotheses can become the final selection. MLESAC replaces the loss function with the amount of inliers, but the score from the loss function is also proportional to the inlier amount. The SPRT test in Cov-RANSAC is a optimal randomized testing, which makes Cov-RANSAC perform slightly better in accuracy compared the "Standard" approach and MLESAC. However, the SPRT relies on the inlier ratio in the sequential hypothesis testing, and the inlier ratio is relevant to the amount of inliers.

TABLE I
EXPERIMENTAL RESULTS OF 1000 SYNTHETIC DATA.

Method	Avg(Δq)	Std(Δq)	Avg(Δt)	Std(Δt)
RCME	**0.010**	**0.013**	**0.534**	**0.665**
pRCME	0.017	0.023	0.720	0.739
Cov-RANSAC	0.139	0.392	0.660	0.810
MLESAC	0.190	0.453	0.741	0.830
Standard	0.193	0.459	0.739	0.838

C. Tests on Public Datasets

We test our method with a wide range of public datasets including one outdoor (KITTI odometry dataset [16]) and two indoor datasets (EuRoC MAV dataset [3] and HRBB4 [23]), as described below.

- KITTI odometry dataset: a outdoor dataset which is recorded by a moving vehicle while driving in Karlsruhe, Germany. We perform tests on 11 image sequences of the KITTI odometry dataset.
- EuRoC MAV: an indoor dataset collected by a synchronized stereo camera. Since this paper only concerns monocular vision, we only utilize the images from the left camera. The EuRoC MAV dataset has 11 sequences, and covers from office rooms to industrial halls with different objects, camera motions and lighting conditions.
- HRBB4: an indoor dataset recored in HRBB 4th floor in Texas A&M University. The HRBB4 dataset is recorded in the office corridor environment.

In all cases, the sequence of two-view image pairs of each dataset are initially selected by applying the initialization module (i.e. key frame selection) of ORB-SLAM2 [25] and we only keep the non-degenerated two-view image pairs. According to different maximum iteration number N settings in the initialization module of ORB-SLAM2, we obtain different sequences of two-view image pairs of each dataset. For each two-view image pair, we use SIFT [22] to obtain the point correspondences as our inputs.

1) Evaluation metric for robustness: Robustness is measured by failure rate of each algorithm. It is important to recognize a failed case.

For every two-view pair, we use the ratio of the consistent inlier amount before and after performing MLE in model refinement step (Box 4s in Fig. 1(a) and Fig. 1(b)) to identify if the method fails to find a correct solution. The underlying rationale is that a quality solution should cause the number of inliers to increase or at least maintain its inlier set size instead of decreasing inlier set size drastically. Also, an incorrect

solution usually falls into local minima instead of the solution close to the global minima in the optimization process. The local minima causes the solution loses the consistency from its inlier data. It is a common heuristic in vSLAM algorithm to check if the map can be created successfully. ORB-SLAM2 algorithm considers a fail initialization if the consistent inlier amount is insufficient after their full BA.

We employ the Huber robust function $\gamma(\cdot)$ on top of reprojection error (page 617 in [18]) to re-evaluate the consistency of inliers before and after MLE. An inlier remains consistent with its model when the value of the Huber robust function is less than the threshold $\tau^2 = \sigma^2 G_2^{-1}(1-\alpha)$, where G_2 denotes the cumulative χ^2 distribution under 2 DoF and $\alpha = 0.05$ is the significance level. Let n_I and n_I^* be the consistent inlier amount before and after MLE, respectively. We set a ratio threshold $\kappa = 0.5$ to determine that the solution is considered as a failed camera motion recovery when $\frac{n_I^*}{n_I} \leq \kappa$. We collect the overall failure rate for each dataset as the measure for robustness for the dataset.

Note that precision that is represented by residual error in cost function is not the concern here because both standard RANSAC and RCME can output high quality solutions in precision if they do NOT fail. In other words, precision is useless if they fail to find the correct solution.

2) Experimental results: The experimental results with 3 different maximum iteration number settings are shown in Tab. II. For each table, the upper half of the table are results from outdoor datasets and the lower half of table are results from indoor datasets. All algorithms perform worse in indoor datasets than that of outdoor dataset. This is because indoor scenarios are more prone to degeneracy. All algorithms gain similar performance under 3 maximum iteration number settings even when the maximum iteration number increases. RCME can achieve 0% average failure rate in the low iteration number.

For both RCME and pRCME, they have the ability to detect poor quality inputs. In those case, the algorithms do not proceed and directly output failure which are represented as detected PQI columns. The "Failure" columns represent the actual failure computed using the metric in Sec. V-C.1. The "Failure" columns do not include cases in PQI columns. In general, we want the failure rate to be as close to zero as possible. It is clear that both RCME and pRCME outperform the "Standard" approach, MLESAC and Cov-RANSAC in the robustness metric no matter how many complete iterations are. RCME achieves 0.00% average failure rate with 0.02% average PQI rate. Our algorithm have improve the robustness of RANSAC for camera motion estimation.

It is also clear that RCME is better than pRCME, which indicates that our model consistency test with its samples works as expected. Also, the fact that pRCME is better than the other three approaches means that our inlier-quality test works as expected.

It is worth noting that MLESAC also gains good performance with overall 0.01% failure rate. The average failure rate of MLESAC of each maximum iteration number setting is close to the average PQI rate of RCME. For instance, the average PQI rate of RCME is 0.16% and the average failure rate of MLESAC is 0.10% when $N = 200$. It is possible that that the poor inputs causing MLESAC to obtain the false camera motion were detected by RCME. Being able to recognize poor quality inputs is a significant advantage of RCME over MLESAC, because we can use this information to trigger the re-selection of key frames.

TABLE II

EXPERIMENTAL RESULTS WITH MAXIMUM ITERATION NUMBER N.

(a) $N = 200$

	RCME		pRCME		Cov-RANSAC	MLESAC	Standard
Dataset	PQI %	Failure %	PQI %	Failure %	Failure %	Failure %	Failure %
KITTI/00(#2424)	0.00	**0.00**	0.00	**0.00**	0.08	**0.00**	0.29
KITTI/01(#58)	1.72	**0.00**	0.00	**0.00**	1.72	**0.00**	3.45
KITTI/02(#2050)	0.00	**0.00**	0.00	**0.00**	0.05	**0.00**	0.15
KITTI/03(#363)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/04(#86)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/05(#1503)	0.00	**0.00**	0.00	**0.00**	0.07	**0.00**	0.40
KITTI/06(#301)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/07(#567)	0.00	**0.00**	0.00	**0.00**	0.53	**0.00**	0.71
KITTI/08(#1711)	0.00	**0.00**	0.00	**0.00**	0.18	**0.00**	0.41
KITTI/09(#687)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/10(#426)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.47
Avg.	0.01	**0.00**	0.00	**0.00**	0.11	**0.00**	0.30
EuRoC/MH_01(#519)	0.19	**0.00**	0.00	**0.00**	0.58	**0.00**	0.77
EuRoC/MH_02(#459)	0.00	**0.00**	0.00	**0.00**	0.66	**0.00**	0.44
EuRoC/MH_03(#589)	0.00	**0.00**	0.00	**0.00**	0.17	0.28	0.51
EuRoC/MH_04(#359)	0.56	**0.00**	0.28	**0.00**	1.39	**0.00**	2.23
EuRoC/MH_05(#404)	0.25	**0.00**	0.00	**0.00**	0.25	**0.00**	0.50
EuRoC/V1_01(#584)	0.17	**0.00**	0.00	**0.00**	0.68	**0.00**	3.08
EuRoC/V1_02(#407)	0.25	**0.00**	0.25	**0.00**	0.98	0.25	2.95
EuRoC/V1_03(#140)	0.71	**0.00**	0.71	**0.00**	2.14	0.71	3.57
EuRoC/V2_01(#402)	0.00	**0.00**	0.25	**0.00**	0.25	**0.00**	2.49
EuRoC/V2_02(#613)	0.16	**0.00**	0.00	0.16	0.16	0.16	3.26
EuRoC/V2_03(#103)	0.00	**0.00**	0.00	**0.00**	0.97	0.97	2.91
HRBB4th(#315)	0.00	**0.00**	0.00	**0.00**	0.32	**0.00**	**0.00**
Avg.	0.16	**0.00**	0.08	0.02	0.57	0.10	1.78

(b) $N = 500$

	RCME		pRCME		Cov-RANSAC	MLESAC	Standard
Dataset	PQI %	Failure %	PQI %	Failure %	Failure %	Failure %	Failure %
KITTI/00(#2493)	0.00	**0.00**	0.00	**0.00**	0.12	**0.00**	0.25
KITTI/01(#65)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	6.15
KITTI/02(#2079)	0.00	**0.00**	0.00	**0.00**	0.05	**0.00**	0.14
KITTI/03(#386)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/04(#91)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/05(#1529)	0.00	**0.00**	0.00	**0.00**	0.13	**0.00**	0.59
KITTI/06(#328)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.91
KITTI/07(#598)	0.00	**0.00**	0.00	**0.00**	0.33	**0.00**	0.17
KITTI/08(#1795)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.39
KITTI/09(#709)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/10(#437)	0.00	**0.00**	0.00	**0.00**	0.27	**0.00**	**0.00**
Avg.	0.00	**0.00**	0.00	**0.00**	0.09	**0.00**	0.31
EuRoC/MH_01(#539)	0.00	**0.00**	0.00	**0.00**	0.19	**0.00**	0.56
EuRoC/MH_02(#483)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.83
EuRoC/MH_03(#679)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.29
EuRoC/MH_04(#402)	0.00	**0.00**	0.00	**0.00**	0.25	**0.00**	1.74
EuRoC/MH_05(#448)	0.00	**0.00**	0.00	**0.00**	0.90	**0.00**	1.34
EuRoC/V1_01(#598)	0.00	**0.00**	0.00	**0.00**	0.33	**0.00**	2.51
EuRoC/V1_02(#424)	0.24	**0.00**	0.00	0.24	0.94	**0.00**	2.59
EuRoC/V1_03(#155)	0.00	**0.00**	0.00	**0.00**	1.31	**0.00**	3.23
EuRoC/V2_01(#395)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	1.77
EuRoC/V2_02(#654)	0.00	**0.00**	0.00	0.15	1.07	**0.00**	1.99
EuRoC/V2_03(#118)	0.00	**0.00**	0.00	**0.00**	0.85	**0.00**	2.55
HRBB4th(#302)	0.00	**0.00**	0.00	**0.00**	0.33	**0.00**	0.99
Avg.	0.02	**0.00**	0.00	0.04	0.44	**0.00**	1.53

(c) $N = 1000$

	RCME		pRCME		Cov-RANSAC	MLESAC	Standard
Dataset	PQI %	Failure %	PQI %	Failure %	Failure %	Failure %	Failure %
KITTI/00(#2491)	0.00	**0.00**	0.00	**0.00**	0.12	**0.00**	0.40
KITTI/01(#80)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	5.00
KITTI/02(#2080)	0.00	**0.00**	0.00	**0.00**	0.10	**0.00**	0.29
KITTI/03(#394)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/04(#95)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	**0.00**
KITTI/05(#1553)	0.00	**0.00**	0.00	**0.00**	0.06	**0.00**	0.52
KITTI/06(#323)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	1.55
KITTI/07(#592)	0.00	**0.00**	0.00	**0.00**	0.17	**0.00**	0.34
KITTI/08(#1762)	0.00	**0.00**	0.00	**0.00**	0.06	**0.00**	0.34
KITTI/09(#712)	0.00	**0.00**	0.00	**0.00**	**0.00**	**0.00**	0.14
KITTI/10(#435)	0.00	**0.00**	0.00	**0.00**	0.23	**0.00**	0.92
Avg.	0.00	**0.00**	0.00	**0.00**	0.09	**0.00**	0.44
EuRoC/MH_01(#574)	0.00	**0.00**	0.00	**0.00**	0.52	**0.00**	1.74
EuRoC/MH_02(#491)	0.00	**0.00**	0.00	**0.00**	0.20	**0.00**	0.61
EuRoC/MH_03(#708)	0.00	**0.00**	0.00	**0.00**	0.28	**0.00**	0.42
EuRoC/MH_04(#400)	0.00	**0.00**	0.00	**0.00**	0.25	**0.00**	2.00
EuRoC/MH_05(#452)	0.00	**0.00**	0.00	**0.00**	0.67	**0.00**	1.11
EuRoC/V1_01(#589)	0.00	**0.00**	0.00	**0.00**	0.51	**0.00**	1.87
EuRoC/V1_02(#445)	0.00	**0.00**	0.00	**0.00**	1.34	**0.00**	3.15
EuRoC/V1_03(#155)	0.00	**0.00**	0.00	**0.00**	1.30	**0.00**	2.58
EuRoC/V2_01(#407)	0.25	**0.00**	0.25	0.25	0.25	**0.00**	2.70
EuRoC/V2_02(#706)	0.00	**0.00**	0.00	0.14	0.71	**0.00**	3.40
EuRoC/V2_03(#136)	0.00	**0.00**	0.00	**0.00**	0.74	**0.00**	0.74
HRBB4th(#312)	0.00	**0.00**	0.00	**0.00**	1.28	0.32	1.28
Avg.	0.02	**0.00**	0.02	0.04	0.60	0.02	1.83

VI. CONCLUSIONS AND FUTURE WORK

We reported our new robust camera motion estimation algorithm targeted at improving robustness in traditional RANSAC-based approaches. Combining two new developments: model quality test with its samples, and inlier quality test with its model, we are able to consistently reduce failure rate of the existing algorithm, as shown in the experimental results from testing a wide range of indoor and outdoor datasets. In the future, we will look deep into those failed

cases that have not been detected by our algorithm and try to design new tests to further improve the robustness of the entire algorithm. We are in the processing of embedding our estimator into existing open source implementation such as ORB-SLAM2 to share the new developments with others.

ACKNOWLEDGMENT

We thank Y. Xu for his insightful discussion. We are also grateful to H. Cheng, B. Li, A. Kingery, A. Angert, D. Wang, and S. Xie for their inputs and contributions to the Networked Robots Laboratory at Texas A&M University.

REFERENCES

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[3] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[4] David P Capel. An effective bail-out test for ransac consensus scoring. In *BMVC*, 2005.

[5] Ondrej Chum and Jırı Matas. Randomized ransac with td, d test. In *Proc. British Machine Vision Conference*, volume 2, pages 448–457, 2002.

[6] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005.

[7] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, pages 236–243. Springer, 2003.

[8] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005.

[9] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[10] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[11] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[12] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[13] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[15] J-M Frahm and Marc Pollefeys. Ransac for (quasi-) degenerate data (qdegsac). In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 453–460. IEEE, 2006.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[17] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[18] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ Pr, 2003.

[19] Peter J Huber. *Robust statistical procedures*, volume 68. Siam, 1996.

[20] Kenichi Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3):171–189, 1998.

[21] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[23] Y. Lu and D. Song. Visual navigation using heterogeneous landmarks and unsupervised geometric constraints. In *IEEE Transactions on Robotics (T-RO)*, volume 31, pages 736 —- 749, June 2015.

[24] Jiri Matas and Ondrej Chum. Randomized ransac with sequential probability ratio test. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1727–1732. IEEE, 2005.

[25] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[26] Kai Ni, Hailin Jin, and Frank Dellaert. Groupsac: Efficient consensus in the presence of groupings. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2193–2200. IEEE, 2009.

[27] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):0756–777, 2004.

[28] Marc Pollefeys, Frank Verbiest, and Luc Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *European conference on computer vision*, pages 837–851. Springer, 2002.

[29] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: a universal framework for random sample consensus. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):2022–2038, 2012.

[30] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. Exploiting uncertainty in random sample consensus. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2074–2081. IEEE, 2009.

[31] Volker Rodehorst and Olaf Hellwich. Genetic algorithm sample consensus (gasac)-a parallel strategy for robust parameter estimation. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 103–103. IEEE, 2006.

[32] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[34] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011.

[35] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.

[36] Ben J Tordoff and David W Murray. Guided-mlesac: Faster image transform estimation by using matching priors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1523–1535, 2005.

[37] Philip Hilaire Torr, Slawomir J Nasuto, and John Mark Bishop. Napsac: High noise, high dimensional robust estimation-it's in the bag. 2002.

[38] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998.

[39] Philip HS Torr, Andrew W Fitzgibbon, and Andrew Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[40] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000.

[41] Philip HS Torr, Andrew Zisserman, and Stephen J Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer vision and image understanding*, 71(3):312–333, 1998.

[42] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. 2020.