Towards a Standalone Methodology for Robust Algorithms Evaluation: A Case Study in 3D Reconstruction

Samuel Victor Medeiros de Macêdo*, Thiago Souto Maior Cordeiro de Farias*,
Juliane Cristina Botelho de Oliveira Lima*, Judith Kelner* and Eduardo Albuquerque†
*Grupo de Pesquisa em Realidade Virtual e Multimídia
Universidade Federal de Pernambuco, UFPE
Recife, Pernambuco
Email: {svmm,mouse,juliane,jk}@gprt.ufpe.br
†Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia, Goiás
Email: eduardo.ufg@gmail.com

***Abstract*—In the field of 3D reconstruction there are two main challenging tasks that require careful consideration, namely, feature detection and matching. The corresponding automatic process introduces noise resulting from the image capture and spurious features matching. A number of robust algorithms for hypothesis evaluation have been suggested; they would deal with these limitations by removing outliers. Most of these works are merely comparisons to previous algorithms and lack any standalone evaluation. This paper attempts to fill this gap by introducing a novel and robust statistical methodology. It has the advantage of evaluating related algorithms using non-dimensional metrics for fixed and continuous intervals. In addition, the proposed methodology is validated using a proof of concept scenario based on the 3D pose estimation phase in the 3D reconstruction pipeline. The obtained results are very promising and emphasize the methodology's generic nature, clearing the way for its application in a multitude of scenarios, such as computer vision and 3D reconstruction.**

***Keywords*-robust algorithms; standalone methodology; statistical tests; hypothesis testing**

I. Introduction

3D reconstruction from 2D images demands multidisciplinary knowledge from several areas, including image processing, computer vision, geometry, and linear algebra, and also demands knowledge relating to nonlinear systems optimization, among other areas. This broad and specific knowledge is required in all phases of the 3D reconstruction pipeline, from 2D image acquisition, to the end product, which is characterized by a cloud of 3D points and by the parameters of the cameras that make up the scene. An example of a 3D reconstruction pipeline can be seen in Figure1. This pipeline is based on Structure from motion (SfM) which is a classical approach to compute the scene structure and camera motion assuming that this information is unknown [1].

This 3D reconstruction pipeline is composed of the following phases: Image Acquisition and Tracking, Fundamental Matrix Estimation, Pose Estimation, Triangulation, Dense Reconstruction and Texturing of the 3D Reconstruction. The first phase is responsible for image processing, since the view acquisition from an image sequence, and the extraction of features in an image, find the correspondent point in the following image. Once the matching point has been computed, the following phase is used to estimate the Fundamental Matrix that encapsulates the projective geometry between the two views, as defined by the epipolar geometry. The Pose Estimation phase is used to calculate the camera matrix [2].

Besides the camera's calibration parameters, extrinsic parameters (the camera pose formed by camera positioning and orientation) are also recovered with 3D reconstruction. A 3D point is reconstructed by triangulation of corresponding points in each group of images and with each corresponding camera pose. The scene can be sparsely reconstructed if only a few thousand points have had their 3D positions computed, or it can be densely reconstructed (called the Dense Reconstruction phase) if the total 3D points extracted are in the millions. Finally, the next phase is to generate texture from the images and render it into the reconstructed 3D model. Further details can be found in [1].

When real data is used in the 3D reconstruction pipeline, there is an introduction of accumulative errors in each executed stage. It starts with image acquisition, which depends on parameters such as image resolution, camera sensor and illumination. The image being processed may have noise when passed on to the next stage of the pipeline. In the tracking phase, aspects such as feature occlusion, false matchings, and drift, due to areas in the image with poor textures or low significant gradients, can also introduce noise into the features positioning along the tracks. Therefore, once it has been acknowledged that there are errors in the data, which were introduced by acquisition and tracking, a new approach that take those errors into account during the calculation

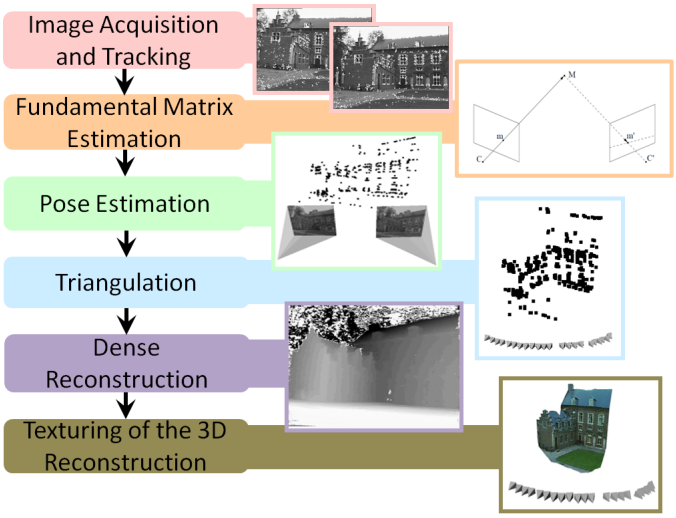

Fig. 1. 3D reconstruction pipeline. Adapted from [1].

of the fundamental, essential and camera pose matrices is required [2].

Since the actual distribution of the error distribution is unknown, it's not possible to filter correct data unless we have the ground truth for the acquisition and tracking stages. Thus, the ground truth can be obtained by methods such as laser scanning, fixed camera path etc., which are expensive techniques, or we can assume that the input data is generated by synthesis. In this paper we will assume that our data is synthetic and that the aggregated noise is modeled with a Gaussian distribution, which is sufficient to infer the robustness of the analyzed algorithm.

To solve the above mentioned problems, a robust estimate of the desired calculus is used to obtain, via random sampling, the model that best adjusts to all observations. Thus, it is possible to separate the correct data according to the selected estimate.

Starting with real images, an uncertainty is added to the data which will generate a hypothesis that corresponds to the estimate object, not to the final product desired (a homography or any other relation).

There are several algorithms available in the literature to generate hypotheses [3], [4], [5] and to evaluate their suitability to the data of the tracking phase. However, the direct application of these algorithms to 3D reconstruction deserves special attention concerning estimate precision. The algorithms were developed and validated comparing their results with other pre-defined algorithms [6], [7], [8], [3]. The above mentioned precision evaluation has been carried out by measuring the re-projection error, which is a recurrent metric in 3D reconstruction. The re-projection error measures how closely the estimates are related to the observed values when the hypotheses are generated.

The comparison of the techniques taking into account only their estimate errors allows the evaluation of final results; however it is not possible to extract important statistics such as undetected outliers. It is also not possible to evaluate how precise the algorithm is, as the error measurement depends on the input data and is influenced by the sample size. The use of an error measurement metric is valid only for direct algorithm comparison, as there is no information about the algorithm precision itself.

Instead of making comparisons among algorithms, as in the existing robust algorithms' analysis, this work proposes a methodology based on robust statistical algorithms to perform hypothesis evaluation, such as RANSAC (Random Sample Consensus) [4] and related works, taking into account the correlation characteristics with the expected response (ground truth) and the presence of outliers. The strong point about the proposed methodology is the introduction of an isolated algorithm evaluation that does not depend on algorithms' comparisons.

Furthermore, the methodology aims to aid researchers interested not only in the re-projection error metric but also in other metrics such as stability, outliers' presence and the correlation level among the selected hypotheses and the ground truth. Another strong point of the methodology is the possibility that the user has to define the tests' precision, regardless of comparisons.

This paper is organized as follows: section II-A introduces the hypothesis to evaluate the algorithms in the context of 3D reconstruction. Section II presents some algorithms that serve as the basis for the implementations available in the literature, and it also presents comparisons among those algorithms. Section III defines the fundamental concepts, presenting the statistical tests and the required analysis to build the proposed evaluation methodology. In section IV a synthetic validation is presented to illustrate the robustness and the practical usage of the proposed methodology. Finally section presents the results and proposes further research directions.

II. State of the Art

The robust estimators' comparison methodologies available in the literature are very simple as to what concerns the comparison metrics. Many works do not perform an evaluation of the estimators but propose a new estimator algorithm which will be analyzing according to a certain metric (usually precision and/or processing time). The comparisons performed by these works are made using algorithms well known in the literature and serve as the basis for other estimators. This section contextualizes some fundamental concepts by describing the most used algorithms in the field, and presents some works on robust estimator comparison.

A. Contextualization

Considering the context of robust evaluation, a hypothesis is commonly generated from a random sample of input data and evaluated in the entire data universe. One hypothesis is a candidate to the final product when it is approved by testing it against a certain threshold. After generating a certain number of hypotheses, the best one, according to a factor defined in the process is seen as the desired result.

This method of hypothesis evaluation is the basis of the RANSAC algorithm [4], an iterative method to estimate parameters of a mathematical model based on a set of data that contains errors. The observations (input data) that do not adhere to a particular set of parameters are called outliers, and those which are faithful to the model are called inliers. The classification of a particular observation as an inlier or outlier is performed by comparing the error generated by the hypothesis test with a threshold tolerance specified by the user. If the error is greater than this threshold, the sample is classified as an outlier.

The original RANSAC uses, as a factor in the evaluation of hypotheses, the number of inliers produced by the evaluation. Thus, a winning hypothesis is always the one with the largest number of inliers.

RANSAC (and its variations) is a method widely used by the community of 3D reconstruction, but it depends on user interaction to determine the test threshold to which the hypotheses will be subjected. There are other methods of evaluation of hypotheses and of threshold definitions that are more appealing by automating calculation and by considering the accumulated error. These methods will be introduced in section II-B.

In the case of the hypothesis used in 3D reconstruction (homographies, fundamental matrix, essential matrices, projection matrices or poses), the error measured for each observation, considering the hypothesis generated in the current iteration, is called reprojection error. This error can be calculated in two ways: first through a simple Euclidean distance between the 2D point (measured by the tracker) and the reprojected 2D point in the image (using a calculated 3D point and a projection matrix); in the second, it uses the distance d between the 2D point (x) and a line in the image (l) (generated from the application of the fundamental or essential matrix to the corresponding point to the given point, present in a different image), where this line is called the epipolar line, as illustrated in Figure 2.

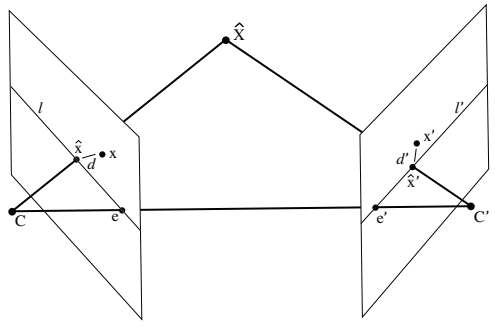

Fig. 2. Epipolar geometry (From [2]). In a pair of images, the projections of an estimated 3D point $\widehat{X}$ must lie on epipolar line, l and l' respectively. The epipolar distance is the sum of the Euclidean distance (d) between x and the point that lie at the foot of the perpendiculars from the measured point to epipolar line $\widehat{x}$, and the equivalent distance (d') on the other image (x' and $\widehat{x}'$).

After calculating the reprojection error and a robust estimate of hypotheses, the observations that are classified as outliers or inliers can be defined. Filtering outliers is very important to discover points that distort the generation of hypotheses, and should not be considered in the process of 3D reconstruction. If any of the outliers is rebuilt, it will not only pollute the generated point cloud but it will also possibly introduce noise in the generation of the mesh that represents it.

B. Robust estimators

The most used robust estimators in the field of 3D reconstruction are based on the RANSAC approach. RANSAC has several parameters that directly influence the choice of the best hypothesis, among them the error threshold that separates the inliers from the outliers, the proportion of inliers in the set (in most cases an assumption), and the probability of a hypothesis containing only inliers. From these parameters, other important factors are also calculated such as the number of iterations of the algorithm, equivalent to the number of hypotheses generated.

Other estimators have emerged with the proposal to reduce the number of parameters present in RANSAC, using the possibility to infer them through a set of input data. Among these is the LMedS (Least Median of Squared) [9], which computes the threshold by estimating the noise present in the data, and follows a Gaussian distribution. There are also estimators that modify the evaluation of hypotheses, using approaches that differ from the classical counting of inliers and the comparison of the re-projection error. In this case, the concept of *M-estimators* was introduced by [5] as estimators that are obtained by minimizing the sum of functions applied to input data. The algorithm that uses the approach of *M-estimator* for the evaluation of hypotheses is called MSAC (M-estimator Sampling Consensus). There are others algorithms which are derived from the MSAC and which use specific *M-estimators* such as MLESAC (Maximum Likelihood Estimator Sampling Consensus) which uses the concept of maximum likelihood [5].

C. Estimator comparison

We did not find in the literature a standard to compare robust estimators.In [6], the adequacy of points to a line in the two-dimensional space was used as hypothesis. This study evaluates only the accuracy of the algorithms using the error normalized square of the inliers [10] as metric. Other similar work proposes a new algorithm called StaRSaC [3] and its evaluation uses as a metric only the measure of distance between the winner hypothesis and the ground truth.

In [7] authors prioritize the evaluation of algorithms considering criteria like time of application and usage of power computing. The metrics used are: number of inliers, number of hypotheses evaluated, number of hypotheses, and speedup achieved in relation to the reference implementation of the RANSAC. In [6] the authors compare the robust estimators using methods including a minimizing non-linear method. The only evaluated metric is reprojection error.

Furthermore, in [11] the evaluation is concerned to the best rotation and translation that aligns the position and orientation

of one data set to the other is constructed by solving an optimization problem. And a statistical method that identifies outliers in the data sets is proposed. But, the results are just compared against other approaches. In [12] the parameters are robustly estimated and the probability distribution of the estimated parameters is evaluated to identify outliers, but there is not a formal approach to evaluate the robustness of the algorithms standalone, neither extensive text to verify algorithm's behavior.

III. Methodology

Considering the lack of a standalone methodology to evaluate robust algorithms, including those for 3D reconstruction, this paper proposes a methodology to fill in this gap.

To use this methodology, the basic requirements are standard (Ω^*) ground truth and the set of vectors produced by the algorithm to be analyzed. These vectors, in the context of 3D reconstruction, can be exemplified as poses (six degrees of freedom floating point vectors, with three degrees of angles and three degrees of translation), fundamental matrices (nine degrees of freedom floating point vectors, one for each matrix element), homographies, and others. In the scenario to be validated in section IV, it will be presented as an evaluation of algorithms for robust estimation of camera poses.

The proposed methodology is defined in nine steps:

1) **Set evaluation scenario**. Choose the application on which the methodology will be used; for example, select from 3D reconstruction, or adjustment of lines, among others, and define what will be the hypothesis.
2) **Generate the ground truth**. After choosing the scenario and the associated hypothesis, a synthetic model should be built (the ground truth).
3) **Generate data entry error associated with an average of μ and variance σ^2 and add outliers**. From the ground truth introduce errors to simulate real world environments. Errors can follow any probability distribution satisfying the conditions of the chosen scenario.
4) **Run the algorithm with the generated data**. Select the algorithm to run and generate the data using as input the sample created in the previous step. The simulation should be run several times according to the Monte Carlo method.
5) **Prepare the solutions so that the variables are independent**. The solutions will be manipulated to ensure the assumptions defined by the tests described in section III-A and III-B.
6) **Set the test parameters**. In this step, the level of rigor of the tests to be applied should be defined.
7) **Apply Pearson test**. At this point in the methodology, there should be verification as to whether the solution is valid or not according to the parameters defined in the previous step.
8) **Select the valid solutions and apply the KS test**. If Pearson's test did not reject the solution, this step will apply the KS test in order to refine the model. This step is essential for obtaining good results and to calculate the coefficient of robustness in the next step.
9) **Compute the percentage of valid solutions and the coefficient of robustness)**. In this last step, the percentage of valid solutions is computed and also the coefficient of robustness of the algorithm according to parameters set in step 6, is obtained.

Considering that the proposed method needs validation, the following sections present the background for the definition of the tests to be performed. First the linear coefficient of Pearson will be introduced and, subsequently, the Kolmogorov-Smirnov test will be described. Concluding this section, an interpretation of the results will be presented.

A. Pearson linear coefficient

Pearson's linear coefficient (PLC) [13], also known as Normalized Cross Correlation (NCC), is a correlation measure between two vectors.

Given a pair of vectors $X_1, ..., X_n$ and $Y_1, .., Y_n$ independent and independent among each other , PLC is calculated as seen in equation 1:

$$r = \frac{\sum_{k=1}^{n}(X_k - \overline{X})(Y_k - \overline{Y})}{(\sum_{k=1}^{n}(X_k - \overline{X})^2)(\sum_{k=1}^{n}(Y_k - \overline{Y})^2)} \quad (1)$$

where $\overline{X} = \frac{\sum_{k=1}^{n} X_k}{n}$ and $\overline{Y} = \frac{\sum_{k=1}^{n} X_k}{n}$. PLC is a well known estimator in the literature; it is effective, easy to implement and interpret, but it has some limitations. The main one is that the relationship between the vectors has to be linear (which is irrelevant to this methodology because, as the aim is to compare the solutions that are similar to the ground truth, the graph taking abscissas and coordinates from the two vectors in question should produce a result close to a straight line with a 45 angle).

Another limitation is that the PLC takes into account the mean vectors. In this case, if the distribution of vectors is asymmetrical or with heavy tails, the average may be inconsistent and therefore the value of r may not be suitable. For purposes of future reference, this work is defined as the Pearson test to the possible rejection of a value r. It rejects the hypothesis of correlation $\forall r$ such that $r < \beta$, where β is the stipulated correlation level.

B. Kolmogorov-Smirnov test

The Kolmogorov-Smirnov (KS) [13] is a nonparametric statistical goodness-of-fit test used to infer the degree of similarity between two probability distributions. In this case the objective is to test the null hypothesis H_0:

$$H_0 : F(t) = G(t) \; \forall t \quad (2)$$

Let $X_1, ..., X_n$ and $Y_1, ..., Y_n$ be two random variables from a continuous distribution, where X and Y are mutually independent and identically distributed. To compute the KS test the empirical distributions $F_m(t)$ and $G_m(t)$ for X and Y respectively should be obtained. The test statistic is given by:

$$J = \frac{mn}{d} \max\{|F_m(t) - G_n(t)|\} \quad (3)$$

where m is the number of observations for the sample X, n is the number of observations to sample and d is the greatest common divisor of m and n. For the decision rule with a level α of significance, H_0 is rejected if $J \geq j_\alpha$, otherwise it is not rejected.

The advantage of using this test is that, as a nonparametric test, you can use any probability distribution and it can therefore be applied in all cases that meet the above assumptions.

It is worth noting that the goal of this test is to compare probability distributions of two samples and not the degree of similarity itself, so when H_0 is not rejected, we highlight that the hypotheses each have a similar probabilistic behavior. Moreover, as shown in equation (3) the statistic is calculated based on $\max\{|F_m(t) - G_n(t)|\}$, so this test is effective in detecting outliers and this is the key point for choosing it (see section IV).

Finally, it should be noted that the KS test does not behave well for constants' vectors, e.g. in case of camera paths (section IV) that lead to little movement in a given coordinate (translation or rotation).

C. Considerations on the methodology

Being λ and γ the results of tests KS and Pearson, the thresholds for evaluating algorithm should be set. For both tests, the closer to 1 the thresholds are, the more rigid is the test and the solution generated by the algorithm is more accurate.

Even though the PLC is a robust metric, the KS test should be additionally performed. This test adds robustness since in a comparison between the probability distributions of the values of the solutions, it is possible to cover differences that point out the presence of outliers. A measure of robustness of the evaluated algorithm can be extracted from the discrepancy between the Pearson and KS tests, since, when an estimate is accurate, both tests should not reject the solution. Thus, there have been three possible outcomes:

- Values above the thresholds for PLC and KS
- Values below the thresholds for PLC
- Values above the thresholds for PLC and below for KS.

When the two tests reach values above the threshold, the algorithm produced good results. Similarly when the Pearson presented values below the threshold the algorithm did not generate a good result. Examples of interpretations will be shown in section IV. These results show a consistency between the tests, independent of the thresholds adopted.

When the solutions have values above the threshold for PLC and below for KS, the generated discrepancy points to a fault in the robustness of the algorithm, which maintains an average sufficiently precise not to be dismissed in the Pearson test; however its probability distribution (empirical) indicates the possibility of the presence of wrong elements within the solution (outliers). Thus, the metric coefficient of robustness (CR) indicated by this work is defined by:

$$CR = 1 - \frac{\text{discordant hypotheses}}{\text{hypotheses accepted by the Pearson test}} \quad (4)$$

IV. Proof of concept

The error distribution is not predictable for real data and therefore it is not possible to filter outliers except under the presence of the ground truth. Since the acquisition of the ground truth is expensive because it depends on a laser scanning, fixed camera path etc., a synthetic test scenario was created in the context of 3D reconstruction using multiple images to validate the steps defined by the proposed methodology. The 3D reconstruction pipeline is divided into several stages as illustrated in Figure 1. The evaluation scenario takes into account, for this proof of concept, the robust pose estimation phase that deals with the errors introduced in the input data from the earlier stages of the pipeline.

The statistical evaluation of the robust algorithms was done through an analysis of the computation of the pose based on the methodology defined, using the implementation of two robust algorithms. The first was the StaRSaC [3] which consists of a method to estimate the parameters that incorporate some features that add value to RANSAC with respect to the computation of an uncertainty threshold. StaRSaC generates a stable solution according to the calculation of the variance of the estimated parameters. Various inliers' acceptance thresholds are generated using an exponential function and for each threshold several hypotheses are generated. The solution is chosen for its adequacy to the model (lower re-projection error and a larger number of inliers), considering the threshold whose generated hypothesis has the smallest variance of the estimated parameters.

The second implemented algorithm was LMedS [9], , which uses the same approach as RANSAC to evaluate hypotheses. In this algorithm, a robust standard deviation is calculated that is used in defining the acceptance threshold of inliers. The calculation of the standard deviation is based on the knowledge generated by the re-projection error of all hypotheses tested. Therefore, this algorithm first generates all the hypotheses, then accumulates the residual errors of each one and afterwards puts them in a vector to allow extraction of their median. Once this step is accomplished, the inliers are defined according to the median obtained. The hypothesis adopted is the one with less residue accumulated.

According to the second step of the methodology, a ground truth of the camera poses was established, where was defined the synthetic camera with a considering as focal length of 1400 pixels. Furthermore, the generated images have a resolution of 800x600 and define as the central point $C = (400, 300)$ with measures also in pixels. The matrix of the intrinsic parameters K is then defined by:

$$K = \begin{bmatrix} 1400 & 0 & 400 \\ 0 & 1400 & 300 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Once the synthetic camera to generate the test scenario was set, 28 poses for cameras with random variation in the three axes of rotation were calculated, and in three dimensions, keeping the camera directed at the scene. The angles were generated randomly within 7 to 13 degrees on the axis of x and y, and from 0.25 to 0.52 in the z-axis, while the translation varied with the distance from the camera to the object which was defined as 100 meters. The trajectory of the camera was directed to move around the object capturing the scene from different angles, as shown in Figure 3, thus obtaining a suitable setting to track features.

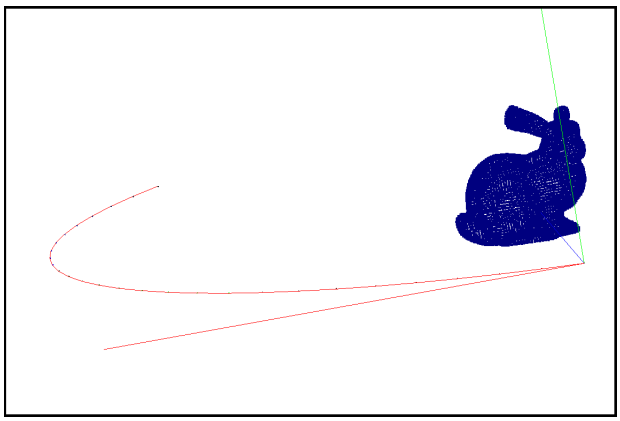

Fig. 3. Camera Path.

The model in this scenario was obtained synthetically from the Bunny 3D point cloud model available in the Stanford repository [14]. Each generated projection matrix was applied to the model, and thus calculated the 2D projection pose for each camera, resulting in 28 respective images. The matches of features among these images were computed by a 3D reconstruction process similar to that described in [1].

Considering that this data set is synthetic, the noise inherent to real images is not captured, such as inaccuracies in the calculation of 2D points and error in the computation of the correspondences between points in different images. To simulate a real environment (step 3 of the methodology) a Gaussian noise in the coordinates of 2D points was inserted and two data sets were generated where the errors follow a normal distribution with standard deviation of 0.5 and 1.0 pixels, respectively. Also, 20

Following the flow of the proposed methodology, the next step (4) was to run the 3D reconstruction pipeline in order to capture the poses generated during the estimation of poses; see Figure 1. A Monte Carlo simulation [15] was performed for each algorithm tested with 1000 repetitions and different samples for each iteration.

Considering that the pose matrix represents a linear transformation on $\mathbb{R}^3$ which groups rotation and translation, for a given sequence of poses, we have the following set of vectors:

$$\hat{\Omega} = \{R_x\, R_y\, R_z\, T_x\, T_y\, T_z\} \tag{6}$$

where R_x, R_y , R_z , T_x, T_y and T_z are rotations and translations on their axes along the path of the camera. Each of these vectors has size i, where i is equal to the number of poses. These vectors are created for the ground truth and for the sample generated by the algorithm to be evaluated according to step (5) of the methodology.

The behavior of the algorithms was evaluated for values of PLC 0.80 (80% similarity with the ground truth) to values of 0.95, which provides a rigid test. For the KS test, we defined the threshold of 0.1 and 0.2, where the first value refers to a test for well behaved data and the second, a test with high rigidity, as specified in step (6) of the proposed methodology.

The next step (7) towards the evaluation of the selected algorithm is the calculation of the linear correlation coefficient of Pearson (PLC) for each vector. Based on this coefficient, the vector that has a coefficient value γ is chosen such that $\forall\hat{\omega} \in \hat{\Omega}$, $\gamma = \min(PLC(\hat{\omega}, \omega^*))$. This ensures that for any element of $\hat{\Omega}$ the Pearson correlation coefficient is always greater than or equal to γ.

Similarly, the Kolmogorov-Smirnov test (step 8 of the methodology) is used in the solutions that are not rejected by the Pearson test and the chosen vector is the one that has a p-value λ such that $\forall\hat{\omega} \in \hat{\Omega}$, $\lambda = \min(KS(\hat{\omega}, \omega^*))$.

The last step of the methodology (step 9) is to compute the percentage of valid solutions and the coefficient of robustness, which are illustrated by Tables I and II.

The first result is the correlation level between the ground truth and the hypotheses. Figure 4 illustrates an example of a valid camera path reconstruction compared with the ground truth. According to the proposed methodology it can be concluded that, for an error of 0.5 pixels, the StaRSaC algorithm produces valid camera paths valid paths cameras for 75.53 % of cases, when dealing with a Pearson test with threshold of 0.85.

Observing Table I, the results for LMeds have the same behavior. It can be noted that LMedS generate 89.89 % of valid camera paths with 0.5 pixel error and with a threshold of 0.85 CLP. This demonstrates that LMedS succeeds in almost 90 % of cases using this threshold, and that it is better than StaRSaC for these parameters.

For the scenario that considers an error of 1.0 pixel the performance of the algorithms was reduced, which was expected, and we can conclude that StaRSaC with 0.85 PLC, for example, produces 49.70% of valid poses while LMedS produces only 41.66% of them. It can be proven that with these results LMedS performs better for samples that show levels of error of 0.5 pixel in the 2D points, and is more effective than StaRSaC for these parameters.

The test rigidity is defined by Pearson's linear coefficient (higher values mean more correlation), and higher PLC values generate fewer valid paths due to the rigor imposed by the PLC. It can be verified in Table I where the percentage of valid camera paths is decreased, for example, from 77.65% to 64.53%, in the case of StaRSaC with 0.5 pixel error. Furthermore, when LMeds and StaRSaC are compared for PLC = 0.95, with 0.5 pixel error, the LMeds achieves the better result, while for 1.0 pixel error they have similar behavior.

TABLE I
PERCENTAGE OF VALID CAMERA PATHS USING PEARSON'S TEST

PLC value	0.80	0.85	0.90	0.95
% of valid camera paths with 0.5 pixel error(**StaRSaC**)	77.65	75.53	71.80	64.53
% of valid camera paths with 0.5 pixel error(**LMedS**)	90.23	89.89	88.69	83.73
% of valid camera paths with 1.0 pixel error(**StaRSaC**)	54.05	49.70	42.17	26.13
% of valid camera paths with 1.0 pixel error(**LMedS**)	45.63	41.66	37.50	25.90

The Pearson test is sensitive to the presence of outliers because the average may not be consistent if the vector distribution is asymmetric or with heavy tails. Thus, because of this misbehavior, another metric is necessary to filter out outliers in a robust fashion. Therefore, when the case is rejected by the PLC test, it must be discarded. Otherwise, the case will be evaluated by the Kolmogorov-Smirnov test, which infers the degree of similarity based on probability distribution.

By using the KS test in the paths of valid cameras, the results in Table II are obtained. The more significant level corresponds to the more rigorous test. As a consequence, there are more rejections when the $KS = 0.2$ than when the $KS = 0.1$. Using 0.5 pixel error, the percentage of rejection by the test $KS = 0.1$ and $PLC = 0.8$ is 3.25, and for $KS = 0.2$ and $PLC = 0.8$ it is increased to 5.30.

Furthermore, when the Person test is more rigid, there are less rejections by the KS test, as can be showed by Table II. It is valid because the approved path cameras are more similar to the ground truth. Considering LMeds result for $KS = 0.10$ with 0.5 pixel error, the percentage of rejection is reduced from 0.85% using $PLC = 0.8$ to no rejection using $PLC = 0.95$.

It can be seen that the LMedS algorithm suffers few rejections compared to StaRSaC, thus proving to be an algorithm that generates softer path poses from the probabilistic point of view. As a result it is concluded that the LMedS generates fewer outliers and is therefore more accurate than StaRSaC, according to the coefficient of robustness defined in equation (4). Figure 5 presents an example of a camera path that was rejected by the tests using $PLC = 0.95$ and $KS = 0.20$ as parameters, then being considered strict testing. Note on the same figure that although the camera path appears to be a visually pleasing alternative, the strictness of tests did not validate it.

After the completion of the test suite for each of the algorithms, this work suggests using a PLC of 0.85, but the user can adjust this threshold depending on the model. As noted in section III, the proposed methodology is relevant considering that one can infer an estimate of the behavior of an algorithm through the input data of the simulation. This methodology introduces a new concept because it does not depend on comparisons between algorithms to verify how suitable the result is for the application, i.e. it is standalone. However, comparisons can be made using the metrics proposed in this methodology as described below.

TABLE II
PERCENTAGE OF CAMERA PATHS REJECTED BY THE KS TEST

PLC Value	0.80	0.85	0.90	0.95
% Rejected by the test KS(0.10) and 0.5 pixel error((**StaRSaC**)	3.25	2.56	1.88	0.51
% Rejected by the test KS(0.10) and 0.5 pixel error((**LMedS**)	0.85	0.85	0.68	0.00
% Rejected by the test KS(0.20) and 0.5 pixel error((**StaRSaC**)	5.30	4.28	3.25	1.54
% Rejected by the test KS(0.20) and 0.5 pixel error((**LMedS**)	2.05	2.05	1.71	0.68
% Rejected by the test KS(0.10) and 1.0pixel error(**StaRSaC**)	4.36	3.57	2.18	0.79
% Rejected by the test KS(0.10) and 1.0 pixel error((**LMedS**)	2.38	1.19	0.79	0.19
% Rejected by the test KS(0.20) and 1.0 pixel error((**StaRSaC**)	11.50	8.73	6.74	2.38
% Rejected by the test KS(0.20) and 1.0 pixel error((**LMedS**)	4.16	2.57	1.38	0.59

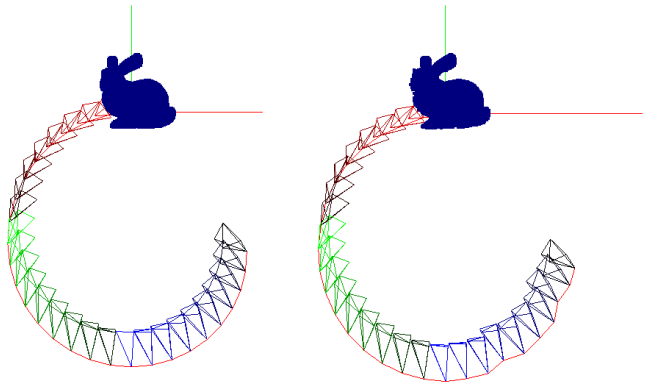

Fig. 4. Left a camera path reconstruction of a sample without noise. On the right a reconstruction with camera path considered valid by the metrics.

V. CONCLUSION

This paper proposed a methodology to evaluate the robustness of a standalone algorithm. The advantage of this technique lies in the simulation and reliability of the proposed statistical tests, without the need for comparison among algorithms, once the suggested metrics are dimensionless and defined within a fixed interval. When real data is used and a ground truth is available, a comparison between these two is natural but a simple comparison between the two results does not offer a statistical evaluation. This ground truth is related to an input image which is set or to fixed image correspondences (without the presence of noise). The raw input data is used to generate some input data with noise, according to a known probability distribution. This generated data feeds the algorithm, which generates several outputs that will be tested against the ground truth through a set of statistical metrics. It is therefore possible to gain some insight on the algorithms robustness. Thus, the actual value of the PLC represents a rank for the solution generated by the algorithm. This would not be achievable considering only

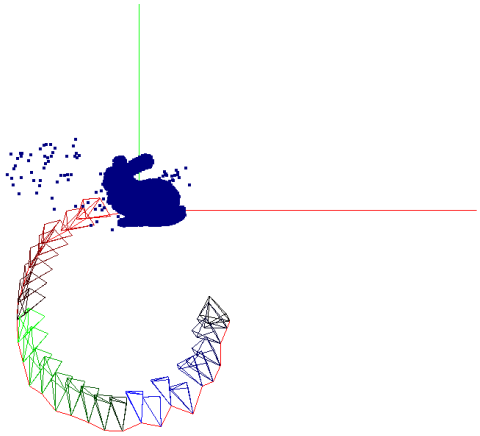

Fig. 5. A reconstruction of the camera path considered not valid for 0.95 and KS 0.20. The points around the object are the result of the triangulation of correspondent points with a high error, so these points are considered as outliers. So this path is considered not valid because there are some wrong pose, and there are a lot of outlier.

traditional metrics related to the re-projection error, since this error is not a dimensionless measure and is subject to the scale in the input data set.

Another contribution of this work is that it defines two metrics: a primary (PLC) and a secondary (KS). These metrics allow the classification of algorithms with respect to the absolute similarity to the ground truth and the possible presence of outliers. The evaluation can be restricted and reliable since it is based on statistical metrics, which are dependent on the level of robustness used in the tests.

As proof of concept of the methodology proposed, it was defined as a scenario and implemented in the 3D reconstruction system, as well as in two robust algorithms in the literature. These algorithms were analyzed and compared according to the proposed metrics. The results were satisfactory and validated the theory described in this study, which confirms that the methodology is a contribution to the field and can be applied in different scenarios.

As future work it is suggested that the methodology be applied to other robust algorithms and other scenarios inherent to the 3D reconstruction context be analyzed, such as the generation of fundamental matrices , essential matrices and homographies. There could also be refinement of the metrics and of the proposed nonparametric tests, which take into account more relevant statistics such as Spearman correlation [13].

REFERENCES

[1] M. Pollefeys, "Self-calibration and metric 3d reconstruction from uncalibrated image sequences," Ph.D. dissertation, ESAT-PSI, 1999.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[3] J. Choi and G. Medioni, "Starsac: Stable random sample consensus for parameter estimation," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 675–682, 2009.

[4] M. A. Fischler and R. C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Morgan Kaufmann Publishers Inc., 1987, pp. 726–740.

[5] P. H. S. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, p. 2000, 2000.

[6] A. J. Lacey, N. Pinitkarn, and N. A. Thacker, "An evaluation of the performance of ransac algorithms for stereo camera calibration," in *In Proceedings of the British Machine Vision Conference (BMVC)*, 2000.

[7] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 500–513.

[8] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," in *In Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[9] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. New York, NY, USA: John Wiley & Sons, Inc., 1987.

[10] S. Choi and J.-H. Kim, "Robust regression to varying data distribution andits application to landmark-based localization," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, 2008, pp. 3465–3470.

[11] M. Shah, "Comparing two sets of corresponding six degree of freedom data," *Computer Vision and Image Understanding*, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2011.05.007

[12] T. Chaperon, J. Droulez, and G. Thibault, "Reliable camera pose and calibration from a small set of point and line correspondences: A probabilistic approach," *Comput. Vis. Image Underst.*, vol. 115, pp. 576–585, May 2011. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2010.11.018

[13] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics, 1999.

[14] Repositrio, "The stanford 3d scanning repository," preprint (2003), available at `http://graphics.stanford.edu/data/3Dscanrep/`.

[15] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*, 2nd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.