**TELKOMNIKA Indonesian Journal of Electrical Engineering**
Vol.12, No.3, March 2014, pp. 1866 ~ 1874
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4465 

# A Novel Quaternion-Based 2D-3D Registration Algorithm with Line Correspondence

**Xinghang Zhang[1], Yuhang He[2]**
[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China
[2]School of Geodesy and Geomatics, Wuhan University, Wuhan, China
#37 Luoyu Road, Wuhan, China, 430079
Corresponding author, e-mail: abstract_cai@foxmail.com[1], yuhanghe@whu.edu.cn[2]

***Abstract***

*Image's registration includes 2D-2D, 3D-3D and 3D-2D registration. This paper only concentrates on the 2D-3D registration, the image's attitude is represented by a rotation matrix R, while the position is a translation vector T. Traditional approaches mainly focus on points correspondences, and state-of-the-art approaches concentrate on high-order structures, i.e. lines, rectangle, parallelepiped etc. Mathematically, most existing solutions adapt either linear optimization or iteration methods. However, they need the position, attitude initialization, which is not always available in real scene, and they do not guarantee to find global solutions. In this paper, instead of solving these polynomials directly, we introduce a novel approach (say qLR), which treat these multivariate polynomial equations as "monomials" and express R in a quaternion vision, resulting in dramatic decrement of the number of equations. We continue to utilize line correspondence since it commonly appears in real-scene and is easy to extract. Experiments on both simulation and real dataset attest qLR's robustness and efficiency. Overall, our approach shares the following advantages: do not need any initialization and guarantee to find the global optimality if it exists; computational-cost is only linear to the number of measurements; robust to noise and separate the calculation of rotation from translation.*

***Keywords****: 2D-3D registration, line correspondence, qLR*



## 1. Introduction

With the popularization and mushroom of camera, smartphone, and other relevant facilities, countless images have been produced everyday by different people from all over the world. The ever-increasing amount of images have brought us a problem: how to classify, store or manage these images? At the same time, modern technological development enables us to reconstruct various objects' 3D model (like historical architectures, street scene, artificial buildings) by either applying multiview geometry, range scanner [1] or other relevant equipments or technologies. This motivates us to classify, store or manage these images according to their position and attitude with respect to the 3D model they have taken. The process to retrieve the position and attitude is called image registration.

Image registration can be coarsely divided into three categories: 2D-2D registration, 2D-3D registration and 3D-3D registration. 2D-3D registration can be simply understood as: given a set of correspondences between the 2D images and their responding 3D model, and the camera's2 intrinsic parameters, namely {f, p($p_1$, $p_2$)} (f : focal length, p: principle points ), are known, with these information, we have to acquire image's location and its pose, namely {R, T }, with respect to the 3D model.

The whole process roughly consists of four steps: 1) Feature extraction and matching: find the correspondence between 2D image and 3D model. 2) Camera intrinsic-calibration: get the camera's intrinsic parameters. 3) Position and pose computation and estimation: get the {R, T }. 4) Follow-up operation: refine the results and eliminate the outliers.

This paper mainly focuses on the third step. The actual methods used in the third step differs according to the different features extracted in the second step. Previous work has harnessed different types of features: point features, rectangle features, parallelepiped features, line features [2], shape features, shadow features, silhouette features and so on. Point-based registration stayed as the center most previous research, while state-of-the-art researches

*Received August 20, 2013; Revised September 24, 2013; Accepted October 12, 2013*

transferred to high-level structures as listed above. At the same time, "too-high-level" structures raise critical requirements to real scenes that few real scenes satisfy to represent them.On the contrary, those not too-high-level structures, like lines, truly exist in most natural scenes, and avoid point-feature-based registration's excessive abundance and uncertainty. That's why we still choose line features for our qLR algorithm.

In both computer vision and photogrammetry, or other relative subject area, the position is presented as a 3×1 matrix $T=[Tx,Ty,Tz]^T$, the pose is presented as a 3×3 rotation matrix R and R is orthogonal matrix. Formally, our input consist of a set of correspondences {C(M), C(I)}, in which the C(M) and C(I) represent the line features extracted from the 3D model and the 2D image respectively. Our output is {R|T}. Note that the key to qLR's success is that there must be enough overlap between 3D model and the 2D image, that is, both of them must capture the same part of the scene. The challenge of qLR is the retrieval of R since R induces a system of multivariate polynomial equations, and its orthogonality must be fulfilled by introducing extra polynomials multivariate equations. Solving these multivariate polynomials requires either iteration or linearization, which unavoidably influences final solutions'accuracy and sometimes cannot find the right solution. On the contrary, our qLR algorithm can successfully address these drawbacks by presenting R in a quaternion-based parameterization and treating these multivariate polynomial equations as "linear monomial equations". What's more, qLR also do not need any initialization or iteration, these significant characteristics guarantee the its efficiency and accuracy.

The following of this paper is organized as: In the second part, a mathematical framework is briefly introduced before describing our qLR algorithm since this mathematical frame serves as the pillar of qLR. The third part is the core of this paper, in which we try to illustrate our qLR algorithm systematically and comprehensively. In the forth part, a large amount of experiments are executed.

## 2. Mathematical Framework

We begin this part by introducing two brief necessary algebraic geometry concepts that will be used in our following qLR algorithm. we consider a system of n quadratic equations of m variables $x_i$ as the form:

$$f_i = \sum_{j=1}^{m} \sum_{k=j}^{m} a_{ijk} x_i x_j (i = 1...n) \tag{1}$$

Note that the right hand of (1) is homogeneous in xi. If the solution of (2.1) is a singlepoint in Rn, we try to linearize (1) by introducing $x_{ij}=x_i x_j$ and ρ=1.Then (1) can be rewrite as:

$$\sum_{j=1}^{m} \sum_{k=j}^{m} a_{ijk} x_i x_j - f_i p = 0 (i = 1...n) \tag{2}$$

As $x_{ij} = x_{ji}$, the new system monomials consist of variables $x_{ij}$ can be solved by singular decomposition. The number of variables then is m(m+1)/2+1, $\{x_{i,j},\rho | 1 \le i \le j \le m\}$.we continue rewrite (2) as:

$$M \bullet \bar{x} = 0 \tag{3}$$

Where $x=[x_{11},x_{12}...x_{1n},x_{22}...x_{nn},\rho]^T$. M indicates the coefficients of (3). We apply SVD decomposition, if $M = S\Sigma V^T$, define $x \in Ker(M)$, $Ker(M)=span(\{vi\})$, where $\{x_i\}$ are the columns of D corresponding to zero singular value in V. When Ker(M) is one dimensional, we recover x to a scale, and the scale can be calculated by the condition ρ = 1.

If the dimension of Ker(M) is larger than 1 (say N>1), we must go on to dig more inner relationship in (1) and (2). Since $x \in Ker(M)$, according to algebraic theory, there exist real numbers $\{\lambda_i\}$ such that:

$$\bar{x} = \sum_{i=1}^{N} \lambda_i \upsilon_i \tag{4}$$

Actually, the elements in x are not independent, we may find that any two elements in x, like $x_{i,j}$ and $x_{i',j'}$, there product equates to another two elements in x, say $x_{i,j'}$ and $x_{i',j}$, since $x_{i,j}x_{i'j'} = x_i x_j x_{i'} x_{j'} = x_i x_{j'} x_j x_{i'} = x_{ij} x_{i'j}$ .So we can easily get then conclusion that: For any integers {i, j, k, l} and their random permutation {i′, j′, l′, k′}, $x_{i,j}x_{kl} = x_{i'j'} x_{k'l'}$.Substituting individual rows of the right hand of (4) into relations of these results discussed just now, through some algebra, we can safely get the constraints on {λi}

$$\sum_{a=1}^{N} \lambda_{aa} (\upsilon_a^{ij} \upsilon_a^{kl} - \upsilon_a^{i'j'} \upsilon_a^{k'l'}) + \sum_{a=1}^{N} \sum_{b=a+1}^{N} 2\lambda_{aa} (\upsilon_a^{ij} \upsilon_a^{kl} - \upsilon_a^{i'j'} \upsilon_a^{k'l'}) = 0 \tag{5}$$

In (2.5), for any integers a and b, we take the notation $\lambda_{ab} = \lambda_a \lambda_b$, $\lambda_{ab} = \lambda_{ba}$. $v_{aj}$ refers to the row of va corresponding to the variable xij in x. Then we can get a new system of linear equations concerning new variables $\{\lambda_{ab}\}$. the (5) can also be rewritten as:

$$K\bar{\lambda} = 0 \tag{6}$$

K is the matrix of coefficients of (5) and λ is the vector $[\ \lambda_{11} ... \lambda_{ab} ... \lambda_{nn}\ ]^T$, the number of λ is N (N+1)/2. Again we solve (6) by applying SVD decomposition, where $K=S\Sigma V^T$. Note that Ker(K) must be one dimensional since two independent solutions would cause two solutions to (1), which contradicts the original assumption.This means that we can recover λ up to a scale.

We briefly summarize what discussed above and try to present a more formal treatment of our approach. Let HQ(Rn) and HL(Rn) be a set of quadratic and linear equations on Rn respectively. These equations are homogeneous in its variables. Our qLR algorithm linearizes the quadratic system listed in (1) to homogeneous system listed in (2) by applying a map f : $HQ(R^n) \rightarrow HL(R^n)$, where the f is defined as:

$$\begin{cases} ft_i t_j = t_{ij} \\ f(1) = p \end{cases} \tag{7}$$

This linearization increase the dimension of our solution to solution space N≥1 by artificially disambiguating the related quadratic terms. If $V_0$=Ker(M), we treat V0 as an N dimensional affine variety in $R^n$.To recover the single solution to (1), we should impose additional constraints by using the equation $x_{i,j}x_{kl}=x_{i'j'}x_{k'l'}$, where {i,j,k,l} and {i',j',k',l'} are each other's permutation.Note that there are many constraints like this, let $eq_1$ be one such equation, and V ar($eq_1$) is its corresponding algebraic variety in $R^n$.Then we denote $V_1$=V0 □V ar($eq_1$) as a subvariety of V0 defined by Equation (1) and system (2). Given an appropriate sequence of such constraints $\{eq_i\}$, we can obtain a nested sequence of variables V0⊃V1⊃V2...in a dimension decreasing order. We guarantee to arrive at a desired solution since there are more constraints than the dimension of V0 itself.

## 3. qLR Algorithm

### 3.1. Problem Formulation

Let$\{L_i^{3D}=(n_i,P_i)\}$ be the input 3D lines, in which $n_i$ denotes $L_i^{3D}$'s normalized direction and $P_i$ denotes any point in the 3D line $L_i^{3D}$.At the same time, let $\{l_i^{2D}=(P_{begin(}X_b,Y_b),P_{end}(X_e,Y_e))\}$, in which $P_{begin}$ and $P_{end}$ denote the two endpoints of the liD respectively. Given pre-known focal length f , all of these 2D lines can be represented in the camera's frame, that is, $P_A=\{X_b, Y_b, f\}$, $P_B=\{ X_e, Y_e, f\}$.

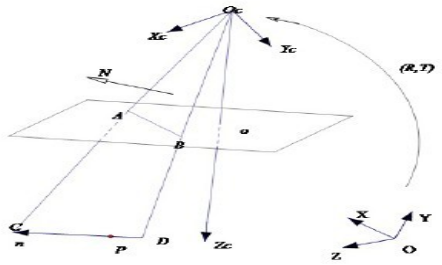


Figure 1. Geometric Constraint on Line-based Registration

According to image-forming principle, 3D model line lCD should 1): lies in the plane P . and 2): after re-projection, any point lies in lCD should be mapped into lADB . we translate these two conclusion into mathematical languages and then get (8) and (9).

$$\vec{n}_i^{\,T} \bullet R \bullet \vec{N} = 0 \tag{8}$$

$$(P^T{}_i \bullet R + T) \bullet \vec{N} = 0 \tag{9}$$

Given a system of measurements, together with a pre-known focal length f, we can calculate a system of (8) and (9) equations. And these equations are bases for the { R, T }'s retrieval.

Notice that the normal vector N of plane P can be directly calculated by focal length f and 2D line $\{l_i^{2D}=(P_{begin}(X_b,Y_b),P_{end}(X_e,Y_e))\}$, et $N_i = [N_{ix}, N_{iy}, N_{iz}]$, then its elements can be calculated in (10).

$$\begin{cases} N_{IX} = Y_E - Y_B \\ N_{IY} = X_B - X_E \\ N_{IZ} = \dfrac{1}{f}(X_E Y_B - X_B Y_E) \end{cases} \tag{10}$$

In real scene, the focal length and the 2D lines, together with their corresponding 3D model lines we received do not share the same measurement. Say, in most cases, the 2D lines and the 3D lines are expressed in pixel, while the focal lengths presented in mm. This force us to transform these data into one measurement. The paper provides us with an available step to achieve this.

### 3.2. Quaternion-based R Representation

As we said in aforementioned sections, qLR derives from three words' initials (Quaternion-based-Line-Registration).We notice that all of these R's expressions suffer from unavoidable drawbacks since they are either computational-cost-high or calculation-difficult. Thus, in this paper, we introduce a novel expression of R, namely Unit Quaternion. Unit Quaternion, also called versers, holds much convenience and significance for representing both orientation and rotation, which are often used to describe 3D objects. Comparing with Euler angles, unit quaternion is not only simple to comprise, but also successfully avoid the problem of gimbal lock. What's more, unit quaternion is numerically stable and efficient.

Unite Quaternion can be expressed as a four-elemental normalized vector $q=[q_0, q_1, q_2, q_3]$, where$q_0^2+q_1^2+q_2^2+q_3^2=1$.Then R can be represented by:

$$R = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{11}$$

Notice that in order to get R, we only need to calculate $q^2$,$q_iq_j$ ,not the $q_i$ itself. that R's orthogonality is inherently-contained and we do not need to add extra constraints to describe it.

Based on (8) (9) and (11), we can construct a system of equations that can retrieve R and T respectively. Given a system of input ∑{ $L_i^{3D}$ ,$l_i^{2D}$ , f }, the constraint equations can be built as $M_{(n+1)11}\times L_{11}=0$, where L = [ $q_0q_1$, $q_0q_2$, $q_0q_3$, $q_1q_2$, $q_1q_3$, $q_2q_3$, $q^0$, $q^1$, $q^2$, $q^3$, 1 ]$^T$the last row of M is always [ 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, −1 ], and the last column of M is always [ 0..., −1 ]$^T$. whatever the measurements are. Actually, other elements in M can be easily calculated as Figure 2.

Since we get M, the SVD decomposition is applied to get the Ker(M), if the dimension of M is 1, the scale of Ker(M) can be recovered by the last row of L. Otherwise, more constraints should be induced to retrieve L. Assume the dimension of Ker(M) is N , that is,

$$L=\begin{bmatrix} q_0q_1 \\ q_0q_2 \\ \vdots \\ q_3^2 \\ 1 \end{bmatrix}=\lambda_1\begin{bmatrix} \xi_{1,1} \\ \xi_{1,2} \\ \vdots \\ \xi_{1,10} \\ \xi_{1,11} \end{bmatrix}+\lambda_1\begin{bmatrix} \xi_{2,1} \\ \xi_{2,2} \\ \vdots \\ \xi_{2,10} \\ \xi_{2,11} \end{bmatrix}+\cdots+\lambda_N\begin{bmatrix} \xi_{N,1} \\ \xi_{N,2} \\ \vdots \\ \xi_{N,10} \\ \xi_{N,11} \end{bmatrix} \tag{12}$$

For any integer { i, j, k, l } ∈{ 0, 1, 2, 3 } and its corresponding permutation { i′, j′, k′, l′}, we could easily get the equation $q_iq_j$ $q_kq_l$ =$q_{i'}$ $q_{j'}$ $q_{k'}$ $q_{l'}$ , and further get $L_{ij}$ $L_{kl}$= $L_{i'j'}L_{k'l'}$, where $L_{mn}$ indicates the element $p_mp_n$ in L.

We then introduce new parameters $\lambda_{ab}=\lambda_a\lambda_b=\lambda_b\lambda_a$, with the constraints expressed in (3.5), we can get another system of linear equations.

$$K_{s\times u}\bullet Q_{u\times 1}=0 \tag{13}$$

Where Q=[$\lambda_1$, $\lambda_1\lambda_2$,..., $\lambda_2$, $\lambda_2\lambda_3$,..., $\lambda_N$ $\lambda_N$]$^T$, the dimension of Q is u =N(N−1)/2+N, the row of matrix K is s=21. We again apply the SVD decomposition to K, and find the Ker(K), the dimension of Ker(K) must be 1 since the dimension that more than one would cause more than one value to Q, which obviously contradicts the proposition that only one single solution exists. In practical experiment, we do not have to calculate all these 21 constraints, since we note that the dimension of matrix U is s ?u, and Ui,i is decreasingly sorted, we sorely care about the member $U_{i,i}$ = 0 ( i = 1,...,N(N−1)/2+N), that is, we can only choose u constraints among all these 21 constraints, which on the one hand let the matrix U be a square matrix, and on the other hand guarantees only the last diagonal element in U is zero.

Since the dimension of Ker(M) is one, the Q is recovered to a scale, we define the scale as λs, all the elements in Q can be expressed with only one coefficient λs. Now, we can square the last row of (12):

$$(\lambda_1\xi_{1,11}+\lambda_2\xi_{2,11}+\cdots\lambda_N\xi_{N,11})^2=1 \tag{14}$$

By solving (14), we can easily get the $\lambda_s$, then the Q is recovered, which means $\lambda_i^2$( i = 1 ???N ) is recovered. the sign of $\lambda_i$ can be decided by re-harnessing the following equations.

$$\begin{aligned}&(\lambda_1\xi_{1,11}+\lambda_2\xi_{2,11}+\cdots\lambda_N\xi_{N,11})^2=1\\&\lambda_i^2\bullet\lambda_j^2=(\lambda_i\bullet\lambda_j)^2\end{aligned} \tag{15}$$

This process of determining the sign of λi is a little tough since many conditions should taken into consideration, especially when the number of λi is high, once λi ( i = 1, ...N ) is retrieved, the R can be solely computed.

### 3.3. Translation Calculation

The calculation of T to merely solve a system of non-homogeneous linear equations [5]. The system of linear equations as Ax=b.· if rank(A) = rank(A| b), there exit no solution; if rank(A) = rank(A| b) = n, there exist one and only one solution; if rank(A) = rank(A| b) < n, there exist infinite solution. If rank(A) = rank(A| b) > n, there exist no solution.

$$A = \begin{bmatrix} N_{1,x} & N_{1,y} & N_{1,z} \\ N_{2,x} & N_{2,y} & N_{2,z} \\ \vdots & \vdots & \vdots \\ N_{n,x} & N_{n,y} & N_{n,z} \end{bmatrix}$$

$$b_{i,1} = [\left\langle p_{i,x}^{3D} \quad p_{i,y}^{3D} \quad p_{i,z}^{3D} \right\rangle \bullet R \bullet \left\langle N_{i,x} \quad N_{i,y} \quad N_{i,z} \right\rangle^{T}]$$

$$x = T = [T_X \quad T_Y \quad T_Z]^T \tag{16}$$

We note that, in most cases, the number of measurements outnumbers 3, which is the least requirements for the accurate calculation of T, since, usually, we can extract a large number of line-correspondences regarding an 3D image and further find their correspondences in the 3D model. Theoretically speaking, there is no solution to these linear equations. However,we can treat this problem as a least-square issue [6], which helps us to find a seemly-ideal solution that minimize the measurement errors.

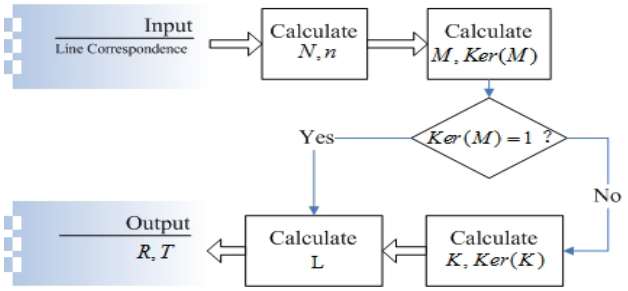


Figure 3. The Main Algorithm Flowchart of our qLR

## 4. Experiments

We take the matlab simulated dataset in [4] as our experimental data and further simulate more data based on [4]. In order to test qLR's robustness and sensitivity to noise, we mainly focus our experiments on three parts

Experiment on the various number of line correspondences.

Experiment on the impact of various degree of noise.

Experiment on the relationship between the degree of noise and the number of line correspondence.

We Let R and T indicate the original rotation and translation respectively and R* and T* indicate the experimental-results (R* for R and T* for T). Then the relative error of R and T is defines as:

$$E_R = \frac{\|R - R^*\|}{\|R\|}$$

$$E_T = \frac{\|R - T^*\|}{\|T\|} \tag{17}$$

Where ‖‖ indicates the F-norm of a matrix.

### 4.1. Number of Line Correspondence

We first use all these dataset free of noise to teat the robustness of our qLR by varying the number of line correspondences.

Table 1. The Relative Error of Registration Based on Different Line Correspondences

| Lines Number | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| R's Error | 0.0690 | 0.0162 | $3.7509\times10^{-6}$ | 0.0226 | $3.7555\times10^{-6}$ |
| T's Error | 0.0690 | 0.0162 | $2.7104\times10^{-9}$ | 0.0226 | $2.4487\times10^{-9}$ |

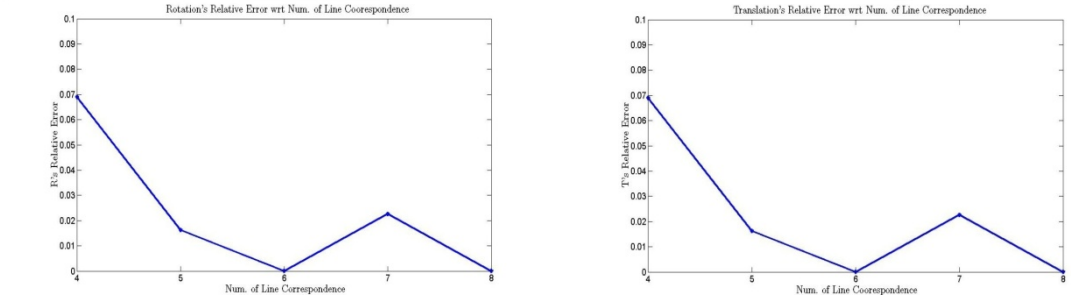


Figure 4. The Relative Error of R and T wrt to the Num. of Line Correspondence that Free of Noise

From the table and the figures, we can conclude that, the accuracy and precision improve dramatically while the number of line correspondences increase. When the number of line correspondences is no less than 4, relative error of both the R and T is within 0.025, which indicates good registration results. Besides, we note that the relative error of R accords with that of T,they share the same trend with the increasing of number of line correspondences.

### 4.2. Tolerance of Noise

In order to testify out algorithm qLR's robustness in the appearance of noise, we deliberately add image noise to the dataset in [4]. Note that the noise added to the image disturbs the normalized normal vector of the projection plane, thus, if the noise level is measured in terms of standard deviation of zero mean Gaussian Distribution, the noise-suffered normal vector N* is:

$$N^* = N + \xi \bullet N_{noise} \tag{18}$$

Where ξ indicates the percentage of noise we would like to add, and Nnoise indicates the original noise we generated. We use five number of measurements for our noise-tolerance experiments, and the results are shown in Figure 5 and Table 2.

Table 2. The Impact of Various Noise Level on qLR

| Noise Level | 0.01% | 0.03% | 0.05% | 0.07% | 0.1% | Noise Free |
|---|---|---|---|---|---|---|
| R's Error | 0.0203 | 0.0585 | 0.0798 | 0.0983 | 0.1244 | 0.0162 |
| T's Error | 0.0104 | 0.0290 | 0.0392 | 0.0460 | 0.0548 | 0.0162 |

From the figure and table, we can safely say that with the disturbance of noise, both the relative error of rotation and translation increase with different speed trend: the R's relative error increase much faster than that of T. At the same time, we note that the increase of R's (the same as T) is linear to the percentage of noise, which indicates that the registration's result is tolerable within a scale of proper percentage of noise.

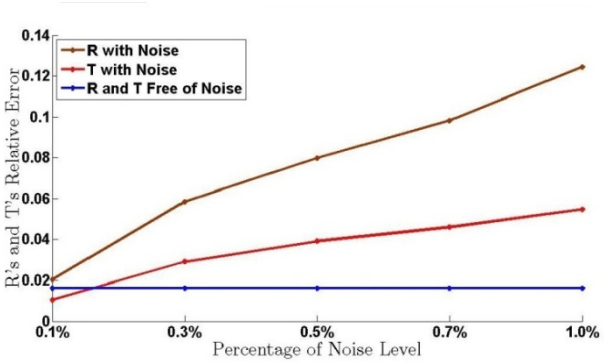


Figure 5. The Impact of Various Random Noise Level on Rotation R and Translation T

### 4.3. Noise Combats with Line Number

The aforementioned experiments show that more number of measurements indicate higher accuracy and precision, while the higher level of noise deteriorates them. Then more exploration on the actual relationship between the number of measurements and the level of noise is desirable and requisite. We try to further the experiment in the noise level section to explore whether the involvement of more measurements would combat with negative impact of noise.

Based on the noise level set above, we extend the number of measurements five to eight and all of them share the same noise level. Table 3 and Figure 6 show our experimental results.

Table 3. Relative Error of Various Noise Level on Eight Measurements

| Noise Level | 0.01% | 0.03% | 0.05% | 0.07% | 0.1% | Noise Free |
|---|---|---|---|---|---|---|
| R's Error | 0.0051 | 0.0156 | 0.0265 | 0.0379 | 0.1415 | $3.7555\times10^{-6}$ |
| T 's Error | $2.8745\times10^{-4}$ | 0.0015 | 0.0037 | 0.0069 | 0.1184 | $2.4487\times10^{-9}$ |

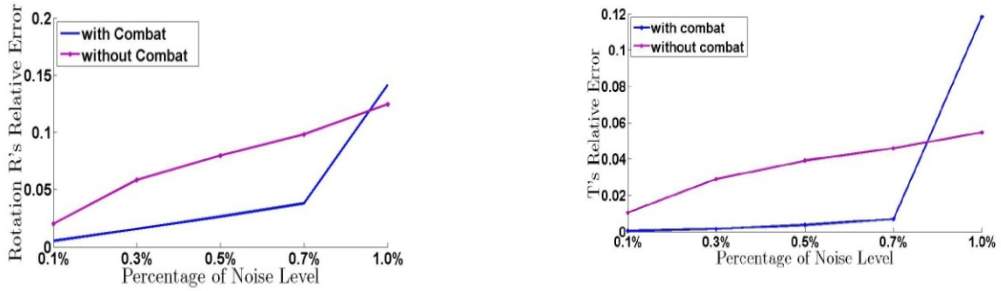


Figure 6. Results of Number of Measurements Combat with Various Noise Level

We can safely get that the involvement of more measurements can significantly soften the negative impacts caused by noise, especially when the noise level is relatively low. Comparing the two figures, we also note that the Translation T benefits more from the add of measurements than that of rotation R, however, the control of relative error caused by involvement of more measurements would eventually shadow its significance when the noise-level is relatively high, thus qLR's robustness mainly depends on its measurements's purity and clearness, rather than on the number of measurements.

## 5. Conclusion and Future Work

In this paper, we have conducted an innovative and efficient algorithm for precisely and accurately registering a camera with respect to its corresponding 3D model. Contrary to

previous relative work, which focusing on either solving a system of nonlinear equations by iterative algorithms, which cannot guarantee to find the global optimal solution or, even worse, a feasible solution, or extracting high-order features, high-requirements of facilities, which do not commonly exist in real scene or is beyond most people's affordability, our qLR formulation reversely accounts for the universality of line correspondence and tactfully treat it as a quaternion-based problem, which enable us to address these multivariate polynomial equations in a "linear monomial equation" vision. Moreover, the complexity of our algorithm is only linear to the number of measurements and it does not require initialization. Various experiments both on simulation dataset and real scene dataset attest the robustness and wide applicability of our algorithm. The appearance of image noise can even be successfully handled.

Note that a system of 2D-3D registration framework consists of several steps, but our algorithm focus on mere one step, this motivates us to concentrate our future work on the whole framework, especially the auto-extraction and auto-matching of line features. Besides, the precise focal length and principle points play a vital role on the registration's results, thus, an accurate intrinsic-parameter-calibration approach is necessary, which also seems to be one of our future work. With the trend of highly intellectualization and modernization, we could easily get the initializations from the smartphone someday. Accordingly, a novel registration algorithm that is extremely robust to noise is requisite, to some extent, we can feel free since we can utilize these initializations.

## References

[1] R Ghazali, A Sukri, A Latif, et al. *Evaluating the relationship between scanning resolution of laser scanner with the accuracy of the 3D model* constructed. Control System, Computing and Engineering (ICCSCE). IEEE International Conference on. 2011.
[2] Mirzaei FM, Roumeliotis SI. *Globally optimal pose estimation from line correspondences*. Robotics and Automation (ICRA). IEEE International Conference on, Shanghai. 2011.
[3] Duraisamy, Prakash, Belkhouche, Yassine et al. 2D-3D registration using intensity gradients. *Signal and Data Processing of Small Targets*, San Diego, CA, United states. 2011.
[4] Liu Y, Huang TS, Faugeras OD. Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE* Transactions on *Pattern Analysis and Machine Intelligence*. 1990; 12(1): 28-37.
[5] Yee-Jin Cheon, Jong-Hwan Kim. Transformation of linear non-homogeneous differential equations of the second order to homogeneous. *Computers and Mathematics with Applications*. 2009.
[6] M. Choi, J. Oh, S Choi. Linearized Recursive Least Square Methods for Real-time Identification of Tire-Road Friction Coefficient. *Vehicular Technology IEEE Transactions on*. 2013; (99).
[7] Hu, Zhangfang, Ji, Chao Luo, Yuan SVD-based mems dynamic testing technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(1): 57-62.
[8] Fan, Shan-Shan, Yang, Xuan. 3D Corresponding control points estimation using mean shift iteration. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 1040-1050.