A Point Cloud Registration Method Based on Point Cloud Region and Application Samples

Yujing Liao*, Fang Xu[2], Xilu Zhao[3], Ichiro Hagiwara*[1]

*Institute for Advanced Study of Mathematical Sciences, Meiji University, Tokyo, Japan

[1]{ihagi}@meiji.ac.jp

[2]Sumitomo Heavy Industries, Ltd.

[3]Department of Mechanical Engineering, Saitama Institute of Technology

Abstract

In this study, a new automatic point cloud registration algorithm based on point cloud registration is proposed to broaden registration ways. The proposed method extracts features of point cloud region for performing the coarse registration. Based on the coarse registration results, the Iterative Closest Point (ICP) algorithm is used for performing the fine registration to restore the measured model. The proposed registration approach is able to do automatic registration without any assumptions about initial positions, and avoid the problems of traditional ICP algorithm in the bad initial estimation. The proposed method along with ICP algorithm provides efficient 3D modeling for computer-aided engineering, computer-aided design and application with Kinect.

1. Introduction

In reverse engineering, the registration for measured data taken from different viewpoints has become a significant research topic [1]. The purpose of registration is to calculate the transforming relation of the overlap part of the measured model for moving measured data into a correct position to restore the model in the cyber place based on measured data from different viewpoints. Conventionally, Iterative Closet point (ICP) algorithm [2] proposed by Besl has been widely used for performing registration.

However, it is very difficult to transform the data to the correct position for restoring the original model using ICP algorithm if measured data from different viewpoints are not close enough. Therefore, when using this algorithm, a modified way is to perform initial registration to move measured data into a close position at first, and then apply ICP algorithm to obtain the correct registration result. In this approach, parameters for performing the initial registration are calculated and determined properly through finding probable corresponding points or combining transforming parameters from measured data.

Furthermore, in recent years, Microsoft Kinect [3] has been used as a 3D Sensor as these inexpensive and commonly available systems can be used to obtain 3D depth information efficiently. The applications for indoor environment by using Kinect with ICP algorithm [4], and researches for indoor map generating of robots by ICP algorithm [5] have become more popular as well.

For using Microsoft Kinect, methods such as [6, 7] are used to perform registration based on ICP algorithm. These methods extract SIFT [8] feature points from color images taken from different viewpoints by Kinect and determine the best corresponding points pairs for calculating the transforming parameters to apply ICP algorithm. However, in some cases such as indoor environment it is very difficult to extract SIFT feature points; consequently, the correct transforming parameters could not be obtained.

In order to solve these new problems brought by recent applications with ICP algorithm, in this study, we propose a point cloud region based method, which could perform the accelerated and correct registration without calculating features of each point. Details of the algorithm are described as follows.

2. Feature extraction

2.1 The significance of analyzing point region

Different from the pre-existing methods [9, 10] based on the feature point extracted through the eigenspace transformation, it is important for our method to analyze the features of the set of points. Hence we proposed a new method based on the eigenspace transformation, analyzing the set of points and extracting features as follows. Here, a set of points from measured point cloud data is called point cloud region, and is denoted as R .

2.2 The central point of the point region *R*

The central point: As the central point of R , the coordinates of p_c can be calculated as follows.

$$p_c = \left\{ \frac{\sum_{i=1}^{N} x_i}{N}, \frac{\sum_{i=1}^{N} y_i}{N}, \frac{\sum_{i=1}^{N} z_i}{N} \right\}. \qquad (1)$$

In the equation N represents the total point number of R , and the coordinates of an arbitrary point p_i of R are denoted as $(x_i, y_i, z_i), i = 1,2,...N$. By using the central point and all points of R , the eigenspace analysis can be performed for this point cloud data. Through the eigenspace transformation, we analyze the point cloud region to obtain more features as follows.

2.3 Analysis by eigenspace transformation

The eigenvector e and the eigenvalue λ : In order to do the eigenspace transformation with all 3D points of R , the Variance-Covariance matrix of R is defined as follows.

$$C = \sum_{i=1}^{N} (p_i - p_c)(p_i - p_c)^T . \quad (2)$$

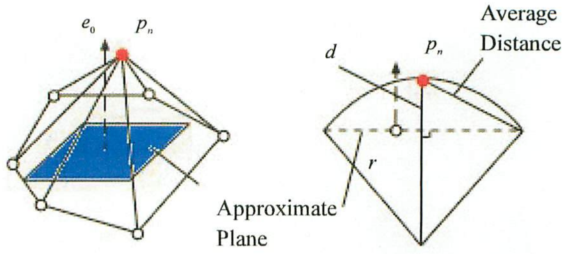
Here, C is the Variance-Covariance matrix of R, $p_i - p_c$ is a 3 times 1 column vector, C is a 3 times 3 matrix. By analyzing the matrix C and calculating the eigenvalue, the eigenvector feature $e\{e_0, e_1, e_2\}$ and the corresponding eigenvalue feature $\lambda\{\lambda_0, \lambda_1, \lambda_2\}(\lambda_0 \le \lambda_1 \le \lambda_2)$ are obtained.

The average distance μ **:** By using the central point p_c and the minimal eigenvalue corresponding eigenvector e_0, the method can be used to generate the approximate plane of all points of R. The distance d between point p_i and the approximate plane, and the average distance μ to the point p_c are calculated as follows respectively.

$$d = \left| e_0^t \cdot (p_i - p_c) \right| . \quad (3)$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \left\| p_i - p_c \right\| . \quad (4)$$

In these equations, $|\ |$ denotes the length of the vector and $\|\ \|$ denotes the distance between two points.



(a) Approximate Plane (b) Calculating Curvature

Fig. 1. Curvature estimation.

The approximate curvature f **:** Search all points of R to find the closest point p_n to the vector that passes through p_c in the direction of e_0. As shown in Fig. 1, the distance, the average distance, and the radius of curvature of the approximate curve surface that consists of all points of satisfy the relationship as follows:

$$\mu^2 - d^2 = r^2 - (r - d)^2 . \quad (5)$$

Here, as the approximate curvature f of R, substituting $r = 1/f$ into (5), f can be calculated as follows.

$$f = \frac{2d}{\mu^2} . \quad (6)$$

The axis direction projective length D **:** Set the eigenvector e_0 as Z axis, e_1 as X axis, and

e_2 as Y axis; all points of R are projected into XY plane. The evaluating value D consists of D_1, D_2 where D_1 is defined as the maximal projected length on X axis whereas D_2 is defined as maximal projected length on Y axis.

The axis direction projective length difference E **:** In positive and negative directions of X axis and Y axis maximal projected lengths are calculated as L_a, L_b, L_c, L_d respectively. The evaluating value E consists of E_1 and E_2, which are defined as follows.

$$E_1 = |L_a - L_b|, E_2 = |L_c - L_d|. \tag{7}$$

2.4 The definition of features

As mentioned above, corresponding features for a point cloud region R are summarized as follows.

$$[p_c \quad e]. \tag{8}$$

$$[\lambda \quad \mu \quad f \quad D \quad E]. \tag{9}$$

Because through the rotating or the translation transformation of R in 3D space, the value of each item in the equation (8) changes as well. p_c is the central point of R, and e is the corresponding eigenvector obtained by performing eigenvalue analysis of R.

In (9) there are invariable features of R, because through the rotating or the translation transformation of R in 3D space, the value of each item in the equation (9) remains the same.

λ is the eigenvalue obtained by performing eigenvalue analysis of R, μ is the average of distances of all points of R to the approximate plane through p_c, f is the approximate curvature of R, and D, E are projection features of on X- and Y- axis respectively.

3. The registration algorithm by using features of point cloud region

The proposed registration algorithm consists of four main parts, and details of the algorithm are described as follows.

3.1 Point cloud region extraction

To solve the problem that we mentioned at the beginning, this study focuses on the possibility of solutions based on the point cloud region. Therefore, here we extracted point cloud regions from measured data by utilizing method discussed in [11].

From point cloud data P, Q that are measured through different viewpoints the extracted the point regions are denoted as follows.

$$\{R_1^P, R_2^P, \ldots, R_i^P, i = 1,2,\ldots,m\}, \{R_1^Q, R_2^Q, \ldots, R_j^Q, j = 1,2,\ldots,n\}. \quad (10)$$

3.2 Matching

3.2.1 The ratio of features

Corresponding invariable feature ratios of a pair of point cloud regions R_i^P, R_j^Q can be calculated as follows.

Table 1. Calculating feature ratios for one pair R_i^P, R_j^Q.

R_i^P	λ_1^P	λ_2^P	λ_3^P	μ^P	...	E_2^P
R_j^Q	λ_1^Q	λ_2^Q	λ_3^Q	μ^Q	...	E_2^Q
Ratio	$RT(\lambda_1^P, \lambda_1^Q)$	$RT(\lambda_2^P, \lambda_2^Q)$	...	...	...	$RT(E_2^P, E_2^Q)$

Here, after values of λ are normalized [10], the ratios of two parameters μ, v are calculated using the function $RT()$ defined as follows.

$$RT(\mu, v) = \frac{Min(\mu, v)}{Max(\mu, v)}. \quad (11)$$

The function $Max()$ is defined as the maximum value of two parameters, whereas the function $Min()$ is defined as the minimum value of two parameters. The sum of the weight of feature ratios for one pair of matched point regions can be calculated as follows.

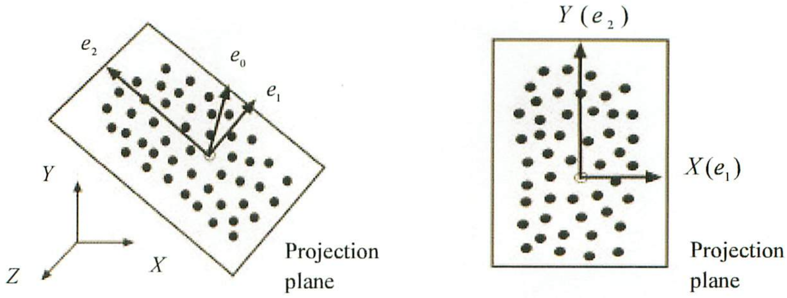
$$k = w_{\lambda_1} \cdot RT(\lambda_1^P, \lambda_1^Q) + w_{\lambda_2} \cdot RT(\lambda_2^P, \lambda_2^Q) \ldots + w_{E_2} \cdot RT(E_2^P, E_2^Q). \quad (12)$$

Weight coefficients for ratio items in the equation are denoted as $w_{\lambda 1}$, $w_{\lambda 2}$, ..., w_{E2} respectively and in order to simplify the calculation they are set equal as $w_{\lambda 1} = w_{\lambda 2} = \ldots = w_{E2} = 1/9$. Furthermore, for any pair of matched point regions R_i^P and R_j^Q, this sum is denoted as k_{ij}. Total number of arbitrary permutation and combination of R_i^P, R_j^Q equals to i times j. Thus, we can rewrite them into a matrix as follows.

$$Matw = \begin{bmatrix} k_{11} & \ldots & k_{1j} \\ \ldots & \ldots & \ldots \\ k_{i1} & \ldots & k_{ij} \end{bmatrix}. \quad (13)$$

The value k_{ij} of a point cloud region pair which has similar features is close to 1; on the contrary, the value k_{ij} of a point cloud region pair which has different features is close to 0. It is possible to judge whether that smaller k_{ij} corresponding of R_i^P, R_j^Q is correct matching pair or not. For this reason, from the matrix we extract bigger k_{ij} with corresponding R_i^P, R_j^Q pairs as follows.

From the matrix for each row the first n' k_{ij} of largest values are extracted with corresponding R_i^P , R_j^Q pairs. Here n' equals to the integral part of δj . The value of δ is determined by experiment and set as 0.55-0.65. For example, when $i=5$, $j=5$, $n'=3$, the three values of the first row of the matrix k_{11} , k_{13} and k_{14} are bigger than the rest of values, corresponding point cloud region pairs (R_1^P, R_1^Q) , (R_1^P, R_3^Q) and (R_1^P, R_4^Q) are extracted. These point cloud region pairs are possible identical area of the measured model.



(a) Projection plane in 3D space (b) Project points into the plane

Fig. 2. Project points from 3D space into 2D plane.

3.2.2 Projection on the approximate plane

Thus if the points of a pair R_i^P , R_j^Q are projected into the same plane, it can be determined that this region pair is incorrect matching pair when the small overlap rate is close to 0.

Furthermore, in order to accelerate the calculation, points on the projection plane are put into different grids. The most left coordinate value and the most bottom coordinate point value from projected points are set as the original point $(0,0)$. The arbitrary projected point coordinate is denoted as (x_i', y_i') . The corresponding number of grids on X axis and Y axis can be calculated as follows.

$$\begin{cases} W = INT(x_i' + 0.5) \cdot N_x \\ H = INT(y_i' + 0.5) \cdot N_y \end{cases} . \tag{14}$$

Here $INT()$ is the rounding function for taking out integer part of the parameter, W represents the number of the corresponding grid on X axis, H represents the number of the corresponding grid on Y axis, N_x is the total grid number on X axis, and N_y is the total grid number on Y axis. Projecting all points of a region pair R_i^P , R_j^Q on the plane, the corresponding grids with projected points are denoted as N_{Pi} , N_{Qj} respectively. The approximate overlap rate of the pair on the plane can then be calculated in (15). In this equation, the intersection of N_{Pi} and N_{Qj} calculates represents the number of overlap grids that have projected points of both point cloud regions, and the union represents the number of overlap grids that have projected points of both point cloud regions, and the union represents the number of grids that have projected points of both point cloud regions. In this equation, the intersection of and represents the number of overlap grids that have projected points of both point cloud regions, and the union represents the number of grids that have projected points of both point cloud regions.

$$Rate_{ij} = \frac{N_{Pi} \cap N_{Qj}}{N_{Pi} \cup N_{Qj}}. \tag{15}$$

This result is then compared with the pre-set threshold value ε . If it is smaller than ε the corresponding pair is determined as the wrong matching pair and then removed. The value of ε is decided by experiment and set as 0.8-0.9. Remaining matching pairs are classified in the next step for calculating the coarse registration parameters.

3.2.3 Multi-group relation

In this step, the remaining point cloud region pairs are classified into different groups through two processes as follows.

Firstly, the central distance ratio is used for classification. For two point cloud region pairs the central distance ratio is defined as follows.

$$s = \left| RT\left(L_P, L_Q\right) \right|. \tag{16}$$

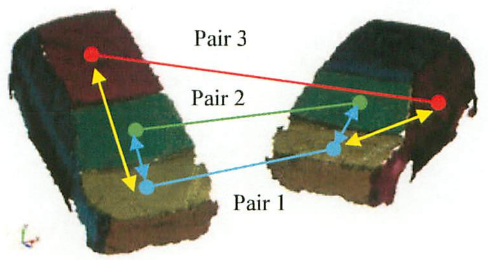


Fig. 3. Point cloud region pairs classification.

Here, L_P is the central distance between central points of two regions R_i^P , R_j^Q and L_Q is the central distance between central points of two regions R_i^Q , R_j^Q from two point cloud region pairs. The function $RT()$ defined previously in (11) is used in the equation. If the value of the distance ratio s in (16) is less than the pre-set threshold ψ , these two point cloud pairs are classified into the same group. The value of ψ is decided by experiment and set as 0.9. As shown in Fig. 3, between pair 1 and pair 2 this ratio is calculated as $s \approx 1$; thus, pair1 and pair 2 are classified into the same group.

Secondly, the included angle is used for classification. As mentioned previously, the eigenvector feature $e(e_0, e_1, e_2)$ which was shown in Fig.2 (a) can indicate the position and the direction of the point cloud region. For this reason, included angles between eigenvector features of the point region pairs from the same group should be close. For example, by using the central distance ratio of point cloud region pairs, (R_1^P, R_3^Q) and (R_2^P, R_3^Q) are classified into the same group, corresponding included angles that between eigenvectors are denoted as $(A0_{13}, A1_{13}, A2_{13})$, and included angles of matching pairs (R_2^P, R_4^Q) are $(A0_{24}, A1_{24}, A2_{24})$. Therefore, the ratio for each corresponding angle from each region that can be calculated as $(RT(A0_{13}, A0_{24}), RT(A1_{13}, A1_{24}), RT(A2_{13}, A2_{24}))$ by using the function $RT()$ in (11). If one ratio is less than the pre-set threshold value, it is determined that the corresponding point cloud pairs are classified incorrectly and should be removed from the group. The value of is decided by experiment and set as 0.8-0.9.

3.2.4 RMS error

In 3.2.3 point cloud pairs are classified into different groups. For each group, by using the central points and eigenvector features of the pairs, transforming matrix is calculated and one time registration for the point cloud P, Q is performed. After that, Root-Mean-Squared error [9, 12] is calculated to evaluate the registration result. The corresponding RMS error is calculated for each group, and the minimum value obtained is denoted as $MinE_{RMS}$. The corresponding RMS error is calculated for each group, and the minimum value obtained is denoted as $MinE_{RMS}$. If $MinE_{RMS}$ is less than a smaller standard convergency value [9], it is determined that the corresponding point cloud pairs of the group are the best matching pairs which can be used for calculating the coarse registration transforming matrix.

3.3 Transforming Matrix

In 3.2.4 the best matching pairs of one group can be obtained. For performing the coarse registration, transforming matrix is calculated based on matching pairs using previous methods in [10] based on the corresponding center point of point regions.

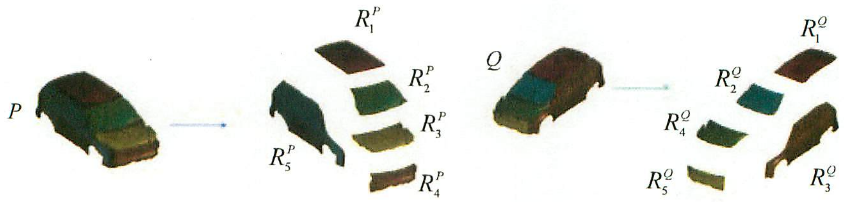
3.4 Transforming

Firstly, by using transforming parameters obtained as above to perform the coarse registration, point cloud data P, Q can be moved into a relatively close position through rotating transforming and the translation transforming.

Secondly, it is fine registration. The second step is fine registration. Based on the coarse registration result and by using ICP algorithm, measured point cloud data P, Q are transformed into a close enough position and the process of the whole registration is finished.

4. Experimental result

As shown on the left side in Fig. 4 (a) and Fig. 4 (b) there are sample data P, Q of a toy car which is measured by a 3D laser measurement machine.



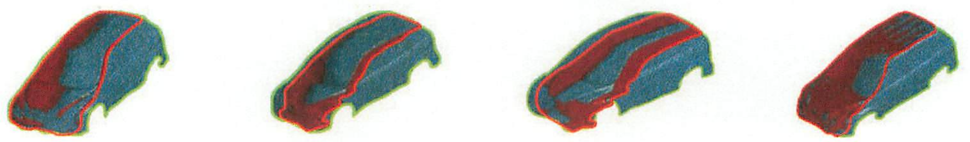
(a) P with corresponding regions (b) Q with corresponding regions

Fig. 4. Point cloud sample data of P , Q with corresponding point cloud regions.

Points of point cloud data P , Q in set 1 are uniform and edges of point cloud regions are very smooth. Points of point cloud data P , Q in set 2 are non-uniform and edges of point cloud regions are not so smooth with some noise. And points of point cloud data P , Q in set 3 are non-uniform and edges of the point cloud regions are not smooth with much noise.

Based on one matched pair in the group obtained above, the transforming matrix is calculated and the coarse registration is performed. Fig. 5 (a), Fig.5 (b) and Fig. 5 (c) show corresponding coarse registration results of three sets data. In Fig. 5, red and green contours represent overlapped parts or separated parts of registration results. By using coarse results, ICP algorithm is applied to perform the fine registration and the result is shown in Fig. 5(d).

The fine registration result is overlapped very well without any separated parts, which is better than coarse registration results The RMS error mentioned in chapter 3.2.4 is calculated for measuring the fine registration result of Fig.5 (d), and the value is 0.217.



(a) Result of set 1 (b) Result of set 2 (c) Result of set 3 (d) Fine registration result

Fig. 5. Coarse registration results and fine registration result by ICP.

(a) The point cloud data P (b) The point cloud data Q (c) Result from the front view

Fig. 6. Application sample by Kinect.

As shown in Fig.6 (a)-Fig.6 (c), there is one indoor application sample taken by Kinect. Point cloud data are shown in Fig.6 (a) and Fig.6 (b). The fine registration result of Fig.6 (a) and Fig.6 (b) is shown in Fig.6 (c), which suggests that the proposed method works well based on the new device.

5. Discussion

As mentioned previously for point cloud regions of non-uniform points and not smooth edges, the proposed method can find correct transforming matrix to move the point cloud data into a close enough initial position for doing the fine registration. Based on these results, the robustness of the proposed method is verified. When applying the proposed method of this research, it is not necessary for the points of the measured point cloud data to be evenly distributed.

For the proposed method, suppose there are N points in the point cloud data, and the number of extracted point regions is N_R; for the toy car sample, $N \approx 4000$ and N_R =10. The calculation time could be reduced as in this method features are extracted from 10 regions whereas in the pre-existing point based method feature for each point ($N \approx 4000$) has to be extracted.

The proposed method can be used when there are few points in the data because point cloud regions can be extracted from the data and there are enough features that can be extracted from these regions.

As the Kinect application sample shows, the point number of measured data for one time becoming $640 \times 480 = 307200$. For this reason the calculation time based on a few of point cloud regions could be reduced compared to the calculation time by pre-existing method 1 which extracts each point feature for over 300,000 measured points. In particular, the proposed method can be applied to the indoor environment where it is difficult to extract feature points from measured data, whereas pre-existing methods based on feature points would not obtain the correct coarse registration result.

6. Conclusion

In this study on the registration in the reverse engineering field, the automatic method based on point regions has been proposed. The robustness of the proposed method is verified and the method could find correct transforming parameters to move the point cloud data into a close enough initial position for doing the fine registration even though point could regions data are not so perfect. The proposed method can be used for the case in which it is difficult to extract feature point but easy to extract point cloud regions, especially for the indoor application based on ICP algorithm.

References

1. Gelfand, N., Ikemoto, L., Rusinkiewicz, S., Levoy, M.: Geometrically stable sampling for the ICP algorithm. In Proc. 2003 The Fourth International Conference on 3D Digital Imaging and Modeling(3DIM 2003), pp.260-267 (2003)
2. Besl, P. J. and McKay, N.D.: A method for registration of 3-D shapes. IEEE Transaction on Pattern Analysis

and Machine Intelligence, vol.14, pp.239-256 (1992)

3. Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: real-time dense surface mapping and tracking. In Proc.2011 IEEE 10th International Symposium on Mixed and Augmented Reality (ISMAR), pp.127-136 (2011)
4. Schindhelm, C. K.: Evaluating SLAM approaches for Microsoft Kinect. In Proc.2011 The Eighth International Conference on Wireless and Mobile Communications (ICWMC 2012), Venice, pp.402-407 (2012)
5. Nüchter, A., Surmann, H., Hertzberg, J.: Automatic model refinement for 3D reconstruction with mobile robots. In Proc.2003 The Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM 2003), pp.394-401 (2003)
6. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research, vol.31, pp.647-663 (2012)
7. Guo, W., Du, T., Zhu, X., Hu, T.: Kinect-based real-time RGB-D image fusion method, International Archives of the Photogrammetry. Remote Sensing and Spatial Information Sciences, XXXIX-B3:XXII, pp.275-279 (2012)
8. Lowe, D. G.: Object recognition from local scale-invariant features. In Proc.1999 International Conference on Computer Vision, pp.1150-1157, Corfu (1999)
9. Xu, F., Zhao, X., Hagiwara, I.: A Study on automatic registration in reverse engineering. Transactions of the Japan Society of Mechanical Engineers, vol.C76, pp.2861-2869 (2010)
10. Xu, F., Zhao, X., Hagiwara, I.: Research on high-speed automatic registration using Composite-Descriptor-Could-Points (CDCP) Model. Transactions of the Japan Society of Mechanical Engineers, vol.A78, pp.783-789 (2012)
11. Curtis, P., Payeur, P.: A method to segment a 3D surface point cloud for selective sensing in robotic exploration. In Proc.2010 IEEE International Workshop on Robotic and Sensors Environments (ROSE), pp.1-6 (2010)
12. Kenny, J. F.: "Root Mean Square" in Mathematics of Statistics. (Pt.1, 3rd ed.), pp. 59-60. Van Nostrand, Princeton (1962)