COMPARATIVE PERFORMANCE EVALUATION OF DIRECTION COSINE MATRIX AND MADGWICK’S AS 3D ORIENTATION ESTIMATION ALGORITHM

IMAM PURWANTO¹, ERI PRASETYO WIBOWO¹,*, DENNIS APRILLA CHRISTIE¹
PURNAWARMAN MUSA¹, ROBBY KURNIAWAN HARAHAP¹ AND RUDI IRAWAN²

¹Doctoral Program in Information Technology
²Mechanical Engineering
Gunadarma University
Jalan Margonda Raya, No. 100, Kota Depok 16424, Indonesia
{imampur; dennis_christie; p_musa; robby_kurniawan; rudi_irawan}@staff.gunadarma.ac.id

*Corresponding author: eri@staff.gunadarma.ac.id

Received September 2020; accepted December 2020

Abstract. 3D orientation estimations are the central point of many scene understanding algorithms including ground contour estimation. The ground contour interval estimation problem can be viewed as a 3D trajectory estimation problem. One framework, called dead reckoning, is often used for motion tracking of a mobile system over time. Dead reckoning requires a precise 3D orientation estimation algorithm. The two most common algorithms are Direction Cosine Matrix and Madgwick’s. To answer whether the 3D trajectory estimation provides accurate results, these two algorithms have been carefully and thoroughly evaluated. The research reported here is part of a complete research on the development of a ground contour estimation. This report focuses only on the comparison between two 3D orientation estimation algorithms.

Keywords: Direction Cosine Matrix, Madgwick’s, Orientation, Trajectory

1. Introduction.

1.1. Background. 3D orientation estimations are the central point of scene understanding algorithms. One of the specific purposes is a ground contour estimation. This paper is part of a complete research of the ground contour estimation algorithm development which focuses on the performance evaluation between two 3D orientation estimation algorithms.

The measurement of ground contour maps is a common activity in geomatics concentration. One piece of the information that can be derived from ground contour maps is the contour interval, i.e., the elevation difference between two adjacent-parallel contour lines [1, 2]. A contour interval is important information that must be provided in a construction development [3, 4]. Unfortunately, a contour interval does not provide any information about the height difference between two contour lines involved, due to the sparsity of the contour lines in the map. The resolution provided also varies depending on its usage.

The problem of ground interval contour estimation (henceforth, will be referred as ground contour estimation), can be viewed from another perspective, namely Three-Dimensional (3D) trajectory estimation of a system. The 3D trajectory estimation is one of the branches in computer science. It is able to produce a collection of 3D point coordinates in space. The 3D point coordinates are able to represent trace or movement of the system while the system is working [5, 6].

DOI: 10.24507/icicel.15.04.409
One of the topics that intersect with this problem is called Dead Reckoning (DR). DR is the movement tracking technique of mobile systems; it tracks relative position and orientation from time to time. The DR is also able to describe the full 3D orientation of a system [7, 8]. However, over time, the error created by the sensor in each measurement moment will accumulate and lead to an exaggerated error [1]. The measurement errors suffered by the DR can actually be minimized by implementing a more accurate orientation estimation algorithm. There are two algorithms that are commonly used in robotics, namely Direct Cosine Matrix (DCM) [2, 7, 9], and Madgwick’s algorithm [10, 11].

Therefore, in order to develop a novel ground contour estimation algorithm with high accuracy, the two 3D orientation estimation algorithms have been compared in this study. The study reported here focused on the implementation and in-depth performance evaluation of the two. The best algorithm would be opted to be part of our complete ground contour estimation algorithm. This paper is organized as follows. Chapter two explains the literature needed in this study. Chapter three describes how the evaluation is conducted. Chapter four shows the results and evaluation. Chapter five concludes the study.

1.2. Problem reformulation. The problems of this study were formulated as follows:
1) How to estimate the 3D orientation of a system?
2) How could the algorithm testing be performed?
3) How could the algorithm be evaluated?

1.3. Objectives. Based on the formulation of the problems above, the objectives of this study were derived as follows:
1) Implementing two 3D orientation estimation algorithms, DCM and Madgwick’s.
2) Using the Inertial Measurement Sensor (IMU) dataset which was available to test whether the algorithm had been implemented properly or not.
3) Determining proper metrics for quantitative evaluation of two involved algorithms.

1.4. Contribution. This paper reports a thorough performance evaluation between orientation estimation algorithms, DCM and Madgwick’s; and summarizes the results which will be useful for future researches that involve this algorithm.

2. Literature Review.

2.1. Orientation representation. Orientation of a system can be described by several representations, including matrix representation and quaternions representation. Each of them has its own unique property requirements and operating rules.

2.1.1. Rotation matrix. A rotation matrix is a matrix that is used to perform a rotation in Euclidean space. It is a $3 \times 3$ matrix in 3D-space. A rotation matrix has two special properties, which are orthogonal and normal.

2.1.2. Quaternions. A quaternion is a 4D number system [12] that extends the definition of complex numbers. Similar to the rotation matrix, quaternions are used to perform a rotation in Euclidean space. However, the quaternions have different sets of rules of operations and properties. A quaternion is represented by a 4D vector of a length one, consists of one and three numbers representing real and imaginary number, respectively.

2.2. Direction Cosine Matrix. The summary of the DCM algorithm is described in Algorithm 1. The details of the algorithm can be found in its original publication [2].

2.3. Madgwick’s algorithm. The summary of the Madgwick’s algorithm is described in Algorithm 2. The details of the algorithm can also be found in their original publications [2, 11].
Algorithm 1: Orientation estimation with DCM

Input:
- Orientation at $t-1$ in rotation matrix $\hat{R}_{(t-1)} \in SO(3)$,
- Current gravitation acceleration $a_t = [a_x, a_y, a_z] \in \mathbb{R}^3$,
- Current angular velocity $\omega_t = [\omega_x, \omega_y, \omega_z] \in \mathbb{R}^3$,
- Current magnetic field $m_t = [m_x, m_y, m_z] \in \mathbb{R}^3$,
- Time interval $dt \in \mathbb{R}$,

Output: Current orientation in rotation matrix $\hat{R}_t \in SO(3)$

Process:
1. Build a skew-symmetric matrix, $[d\theta]_x$, based on gyroscope orientation ($\omega'$).
2. Update the rotation matrix, $R_t$.
3. Normalize the rotation matrix, $\hat{R}$.
4. Detect the error (drift) for roll-pitch $e_{rp}$ and yaw $e_y$.
5. Correct the error with proportional and integral term to obtain corrected gyroscope orientation $\omega'$.

Algorithm 2: Orientation estimation with Madgwick’s

Input:
- Orientation at $t-1$ in Quaternions $q_{t-1} = [w, x, y, z] \in \mathbb{R}^4$, $\|q\| = 1$,
- Current gravitation acceleration $a_t = [a_x, a_y, a_z] \in \mathbb{R}^3$,
- Current angular velocity $\omega_t = [\omega_x, \omega_y, \omega_z] \in \mathbb{R}^3$,
- Current magnetic field $m_t = [m_x, m_y, m_z] \in \mathbb{R}^3$.

Output: Current orientation in Quaternions $q_t = [w, x, y, z] \in \mathbb{R}^4$, $\|q\| = 1$,

Process:
1. Calculate quaternions rate of change, $\dot{q}_t$, based on gyroscope.
2. Calculate quaternions orientation, $q_{\omega,t}$, based on gyroscope.
3. Calculate Loss Function, $f_{g;b}$.
4. Calculate Jacobian Matrix, $J_{g;b}$.
5. Perform Gradient Descent at $q_{t-1}$ to obtain corrected quaternion $q_{\nabla;t}$.
6. Fuse two quaternion measurements ($q_{\omega,t}$ and $q_{\nabla;t}$) to obtain fused quaternion estimation ($\hat{q}_{est;t}$).
7. Normalize quaternions $\hat{q}_{est;t}$ to obtain normalized quaternion estimation $q_{est;t}$.

3. Comparison Framework. To observe the performance of both Algorithm 1 and Algorithm 2, a test using real data was performed. The secondary data, derived from [10], were used as the inputs for both of the algorithms.

The available dataset was a pair of files with CSV (comma separated values) extension. The first file, dataset.csv, was a file that contained raw data from 9 DoF IMU measurements. The dataset.csv file had 6960 data rows including headers. The second file, groundtruth.csv, was a file that contained groundtruth in Euler space. The numbers of rows in groundtruth.csv file were the same as in the dataset.csv file. Table 1 shows the IMU measurement scenario description within the dataset. The scenario was divided into seven stages, arranged in such a way that all IMU’s basic orientations were read.

Performance evaluation and comparison of the two algorithms were completed using MATLAB 2016b software on a computer with Ubuntu Operating System 14.04 LTS with specifications of i5-6200U @ 2.3 GHz CPU and 4 GB RAM.
Table 1. Scenario description within dataset

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The IMU sensor is first placed statically.</td>
</tr>
<tr>
<td>2</td>
<td>The IMU is moved 90° to the X axis, stops, −90° to the X axis, stops.</td>
</tr>
<tr>
<td>3</td>
<td>The IMU is returned to its original position.</td>
</tr>
<tr>
<td>4</td>
<td>The IMU is moved 90° to the Y axis, stopped, −90° to the Y axis, stopped.</td>
</tr>
<tr>
<td>5</td>
<td>The IMU is returned to its original position.</td>
</tr>
<tr>
<td>6</td>
<td>The IMU is moved 90° to the Z axis, stopped, −90° to the Z axis, stopped.</td>
</tr>
<tr>
<td>7</td>
<td>The IMU is returned to its original position.</td>
</tr>
</tbody>
</table>

3.1. Qualitative evaluation. Qualitative evaluation was performed by superimposing the collection of estimated data and groundtruth into the same graph.

If there is a significant difference between the groundtruth orientation line and the estimated orientation line, it indicates that the performance of the algorithm is low. It means that the algorithm is unable to estimate the orientation as the actual situation.

3.2. Quantitative evaluation. To conduct a quantitative evaluation, it is required to calculate the errors of orientation estimation as compared to the groundtruth for each given moment. The error is calculated using the error function shown in Equation (1),

\[
E(Q, \hat{Q}) = \frac{1}{N} \sum_{i=1}^{N} \| Q_i \otimes \hat{Q}_i^{-1} - 1 \|_2
\]

where \( \| \cdot \|_2 \) is the l2-norm, \( 1 \in \mathbb{R}^4 \) is the identity quaternion which has value of 1 for the real components \( (w) \) and 0 for all imaginary components \( (x, y, z) \), \( Q \in \mathbb{R}^4 \) is the ground truth orientation in the quaternion, \( \otimes \) is the symbol of quaternion product, and \( \hat{Q} \in \mathbb{R}^4 \) is the orientation estimation in quaternion with superscript \(-1\) representing the inverse of quaternion that is defined as,

\[
\hat{Q}_i^{-1} = \frac{\hat{Q}_i^*}{\| \hat{Q}_i \|_1}
\]

where \( \hat{Q}^* \) is the quaternion conjugate from \( \hat{Q} \) and \( \| \cdot \|_1 \) is the l1-norm from \( \hat{Q} \).

\( Q_i \otimes \hat{Q}_i^{-1} \) in Equation (1) produces an identity quaternion when the estimation \( \hat{Q} \) predicts the orientation \( Q_i \) correctly with no errors. This is the most optimal condition, so that the difference between \( Q_i \otimes \hat{Q}_i^{-1} \) and the identity quaternion \( 1 \) is zero.

The error function, defined by Equation (1), shows the performance of the corresponding orientation estimation algorithm, i.e., the smaller the value calculated using Equation (1), the more accurate the estimation made by the algorithm; and vice versa. Value range produced by the error function is from zero \((0)\) to one \((1)\).

4. Result and Evaluation.

4.1. Estimated orientation results. The IMU measurement dataset at its 9 DoF was used as the input for Algorithm 1 (DCM) and Algorithm 2 (Madgwick’s). Both of these algorithms produced outputs in the form of collection of estimated orientation in the Euler angle. These data collections are represented in Figures 1 and 2.

The red, green, and blue lines (see Figures 1 and 2) represent the orientation estimation for the X, Y, and Z axis, respectively. It is clearly shown in the graph that there are three main waves that are formed. Each of waves is the representation of conducted stage number 2, 4, and 6 (see Table 1).
4.2. **Qualitative evaluation.** Performance evaluation of Algorithm 1 and Algorithm 2 can be conducted by superimposing the actual orientation (groundtruth) with the estimated orientation in one graph (see Figures 3(a) and 3(b)). Qualitatively, the estimated orientation (red line) almost always coincides with the groundtruth (blue line), particularly in the three main waves mentioned earlier. It indicates that the Algorithm 1 and the Algorithm 2 are able to estimate the orientation of the system correctly.

There are sudden spikes that only happen in the estimated roll (Figure 3 – 1st column) and estimated yaw (Figure 3 – 3rd column). This spike is not found in the estimated pitch (Figure 3 – second column). The spikes do not indicate that Algorithm 1 and 2
fail to estimate the orientation, these are gimbal-lock artefacts which only occur in Euler angle representation. It happened in a specific situation, i.e., when the system orientates itself towards 90° on the Y axis (pitch).

The Euler angle is indeed the most intuitive orientation representation of a system to be analyzed by humans. However, due to the limitation possessed by this representation, the performance of Algorithms 1 and 2 should be done using another orientation representation. Figures 4 and 5 are comparisons between the orientation estimation and the groundtruth in quaternion representation. Red, green, blue, magenta lines are the w, x, y, z components of quaternions, respectively.

The results of qualitative evaluation does not show a significant differences between the two algorithms. There is slight deviation performance, but it is quite difficult to be observed. Hence, a quantitative test needs to be conducted.

4.3. Quantitative evaluation. Quantitative analysis was conducted on the two algorithms. To strengthen the analysis, these algorithms were compared at different noise levels. The baseline noise was estimated by measuring the standard deviation (σ) of the mean-normalized data on each axis (x, y, z) of each sensor (accelerometer, gyroscope, magnetometer). Figure 6 shows the original (red lines) and the noise added measurement (green lines).

Figure 7 shows the comparison of errors of the two algorithms using dataset inputs that are subject to 11 different noise levels, from 0σ to 5σ. The estimated results of each noise level are compared to the groundtruth using the error function (Equation (1)). To ensure quantitative measurements, the two algorithms were run n times (n = 10 in this study).
so as to produce a collection of $n$ error values. The mean of the errors is plotted on the graph and becomes the reference for this quantitative analysis.

DCM algorithm (blue line) tends to have smaller error value than the Madgwick’s algorithm (red line). Therefore, it can be concluded that the DCM algorithm has advantage in resistance to noise.
Figure 5. Comparisons of Madgwick’s orientation estimation with groundtruth using quaternion representation

Figure 8 shows the comparison of execution speed between two algorithms in completing its operations for all available data in the dataset. The duration was measured in seconds. The blue bar indicates the duration of pure operation required by each algorithm, i.e., without any data stored into variables or other things. The green bar indicates the duration of operation with a storing process of the estimated orientation results into a variable. The yellow bar shows the duration of operation with a transforming process...
Figure 6. Original and noise added measurement with $5\sigma$.

Figure 7. Comparisons of DCM and Madgwick’s in resistance to noise from the body-frame coordinate system to the world-frame coordinate system (a process that must be done if the orientation in the observer coordinate needs to be represented).
Madgwick’s algorithm (left) is superior in speed as compared to the DCM algorithm (right). This result is consistent with one of the claims in the original publication of Madgwick’s Algorithm [2].

Figure 9 shows the comparison of the amount of memory required for each algorithm to store the entire collection of estimated system orientation values from the dataset. Memory size is measured in bytes.
It can be concluded that the Madgwick’s algorithm (left) utilizes memory more efficiently as compared to the DCM algorithm (right). This conclusion also agrees with one of the claims in the original publication. This is because Madgwick’s works in quaternion space, which is a four dimensional vector (it requires 4 floating point values) while DCM works using a $3 \times 3$ transformation matrix (it requires 9 floating point values).

The three comparisons are summarized in Table 2, with (+) sign indicating superiority and (−) sign indicating inferiority.

Table 2. Algorithm comparison summary

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>DCM</th>
<th>Madgwick</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error + Noise</td>
<td>(+)</td>
<td>(−)</td>
</tr>
<tr>
<td>Execution duration</td>
<td>(−)</td>
<td>(+)</td>
</tr>
<tr>
<td>Memory utilization efficiency</td>
<td>(−)</td>
<td>(+)</td>
</tr>
</tbody>
</table>

5. **Conclusion and Future Work.** To perform ground contour estimation, one part of the algorithm that needs to be evaluated carefully is the orientation estimation algorithm as it is the key factor to estimate system trajectories (the ground contour). Two algorithms have been evaluated, i.e., 1) Direct Cosine Matrix (DCM), and 2) the Madgwick’s.

The test was conducted using the IMU sensor measurement dataset. Qualitatively, both algorithms are able to estimate orientation based on predetermined IMU measurement data. The differences are not significant, so that quantitative test is required. Based on three quantitative tests, the DCM algorithm is superior in accuracy and resistance to noise, while Madgwick’s algorithm is superior in execution duration and memory efficiency. Each algorithm does have advantages and disadvantages, so the results of the tests conclude:

1) If the available hardware has limited specifications (example: a microcontroller), Madgwick’s algorithm is a better choice,

2) However, if the available hardware has high specifications and is flexible (for example: a computer), the DCM algorithm is a preferred choice.

Because a computer will be used as the machine for applying the algorithm in future research (ground contour estimation), DCM is a wise choice to be the orientation estimation algorithm. The main priority in this research is to produce contour plots with high degree of accuracy. The duration and amount of memory are the second priority because computer specifications can be upgraded easily according to algorithm requirements.

**REFERENCES**


