EE267 3 Unit: Impact of Compute on Orientation Tracking

Munir Bshara
Stanford University
450 Jane Stanford Way, Stanford, CA 94305
bshara@stanford.edu

Abstract

Virtual Reality and Augmented Reality devices face inherent constraints in power consumption and computational resources, yet require accurate 3-DOF orientation tracking for optimal user experience. While more sophisticated algorithms are computationally feasible on these platforms, it remains unclear whether the additional computational overhead translates to meaningful improvements in user experience. This study investigates the relationship between algorithm complexity and user-perceived performance by comparing four orientation tracking algorithms: Extended Kalman Filter (EKF), Madgwick, Mahony, and complementary filter approaches. Through user testing with 16 participants, I evaluated whether computationally intensive algorithms provide superior user experience compared to simpler alternatives on a resource-constrained VR device. Results indicate no significant correlation between algorithmic complexity and user-reported experience quality, suggesting that simpler, more power-efficient algorithms may be preferable for battery-powered immersive devices without sacrificing user satisfaction.

1. Introduction and Motivation

Virtual Reality device design involves critical tradeoffs between performance, power consumption, and user experience. While some challenges like display optimization remain active research areas, orientation tracking represents a well-established domain with multiple proven algorithms ranging from simple complementary filters to complex sensor fusion techniques.

In today's competitive technology landscape, engineers often operate under the assumption that available compute should be fully utilized, gravitating toward sophisticated algorithms when processing power permits. However, this "compute is free" mentality raises a fundamental question: does increased algorithmic complexity actually improve user experience?

This study evaluates whether computationally intensive orientation tracking algorithms provide better user experience compared to simpler alternatives, providing empirical evidence to guide resource allocation decisions in VR system design.

2. Related Works

Orientation tracking algorithms have a rich history in navigation applications [2], where accurate dead reckoning is critical for GPS-denied systems. However, for Virtual Reality technology, which has only emerged as a consumer platform within the past decade, limited research exists examining the relationship between algorithm choice and user experience.

LaValle et al.'s 2014 paper [1] represents one of the few works that explicitly considers how orientation algorithm design impacts VR user experience. While the authors claim their complementary filter approach improves user experience, they provide no methodology for measuring this improvement, nor do they justify why this approach was selected over alternatives such as Extended Kalman Filters, Madgwick, or Mahony filters.

Most existing literature, such as Cocoli et al. [4], evaluates orientation tracking algorithms purely on technical metrics like reprojection error relative to ground truth trajectories. While these approaches provide insights into algorithmic accuracy, they disregard human factors that define VR user experience, failing to account for natural human movement patterns or perceptual factors such as motion sickness that directly impact user comfort.

This gap between technical performance metrics and actual user experience motivates the need for user-centered evaluation methodologies that can inform algorithm selection for consumer VR devices.

3. Experimental Method

To assess different algorithms while ensuring users engage in realistic VR tasks, I developed a controlled experimental environment that mimics typical VR usage patterns.

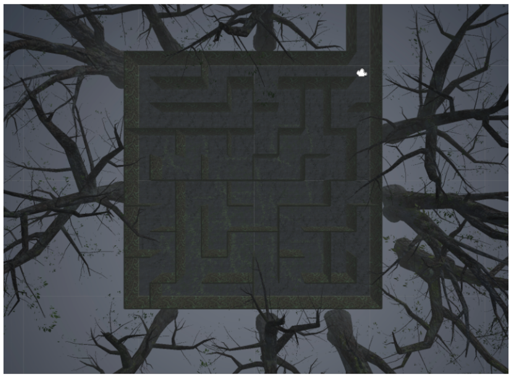

Figure 1. Bird's eye view of Maze users traverse.

Figure 2. Ghost chases you down.

3.1. Game

I created a maze escape game in Unity that challenges users to navigate through a procedurally generated maze while being pursued by a ghost. Orientation tracking is controlled by natural head movement, while translation is controlled using keyboard input (WASD keys). The maze is procedurally generated to prevent learning effects. Figure 1 shows an example maze configuration.

The player and walls have collision hitbox to prevent the user from looking or moving through a wall. The ghost does not and can follow users through walls. The trees and foliage was made using JP Environmental Asset Package, the sky was obtained from AllSkyFree, and the ghost was found on a the official Unity subreddit.

3.2. Algorithms

I evaluated four orientation tracking algorithms using accelerometer and gyroscope data from an IMU. While sensors were not formally calibrated prior to testing, empirical observation confirmed near-expected performance. The algorithms, ordered from least to most computationally intensive, are described below with their quaternion-based implementations:

Linear Complementary Filter:

$$q_c^{(t+\Delta t)} = q\left((1-\alpha)\,\phi,\ \frac{\mathbf{n}}{\|\mathbf{n}\|}\right) q_\omega^{(t+\Delta t)} \tag{1}$$

where α is your complementary gain.

Mahony AHRS Filter(PI based complementary):

$$\omega_{t+1} = \omega_t + K_p e_{t+1} + K_i e_{i,t+1} \tag{2}$$

$$\dot{q}_{\omega,t+1} = \frac{1}{2} q_t \otimes \left[0,\ \omega_{t+1}\right]^T \tag{3}$$

$$q_{t+1} = q_t + \Delta t\, \dot{q}_{\omega,t+1} \tag{4}$$

where e represents the cross product error between estimated and measured gravity, and K_p, K_I are the proportional and integral gains.

Madgwick Filter (Gradient-descent based) [4], [3]:

$$q_{\nabla,t+1} = -\beta \frac{\nabla f}{\|f\|} \tag{5}$$

$$\dot{q}_{\omega,t+1} = \frac{1}{2} \hat{q}_t \otimes \left[0,\ \omega_{t+1}\right]^T \tag{6}$$

$$q_{t+1} = \hat{q}_t + \Delta t\left(\dot{q}_{\omega,t+1} + q_{\nabla,t+1}\right) \tag{19}$$

where β is the step size, $f(q)$ represents the objective function incorporating accelerometer measurements, and ∇f is the jacobian matrix applied on f.

Extended Kalman Filter (Full Probabilistic Estimator) [4]:

$$x_t = q_t = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \omega_t = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad Q_t = \sigma_Q^2 I_4, \quad R_t = \sigma_R^2 I_3.$$

$$\dot{q}_t = \frac{1}{2}\,\Omega(\omega_t)\, q_t, \qquad q^-_{t+\Delta t} = \mathrm{norm}\big(q_t + \dot{q}_t\,\Delta t\big),$$

$$F_t = I_4 + \frac{\Delta t}{2}\,\Omega(\omega_t), \qquad P^-_{t+\Delta t} = F_t P_t F_t^T + Q_t.$$

$$y_t = a_t - h\big(q^-_{t+\Delta t}\big),$$

$$S_t = H_t P^-_{t+\Delta t} H_t^T + R_t, \qquad K_t = P^-_{t+\Delta t} H_t^T S_t^{-1}.$$

$$q_{t+\Delta t} = \mathrm{norm}\big(q^-_{t+\Delta t} + K_t y_t\big), \qquad P_{t+\Delta t} = \big(I_4 - K_t H_t\big)\, P^-_{t+\Delta t}.$$

where $\Omega(\omega_t)$ is the lie algebra generator for $SO(4)$ ie a skew-symmetric matrix that when exponentiated represents a finite rotation, H is known as the Innovation which is just the Jacobian as acquired in equation 4, and I_i is an identity matrix of size i.

Regarding implementation, the complementary filter follows the starter code, the Mahony and Madgwick filters are implemented as described by Çoçoli and Badia [4], and the Extended Kalman Filter (EKF) implementation is adapted from the online AHRS documentation,cite.

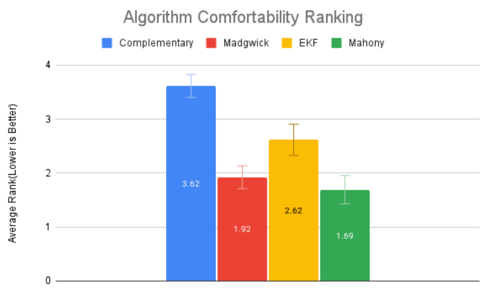

Figure 3. Average comfort ranking versus algorithm, averaged over 14 users (lower is better).

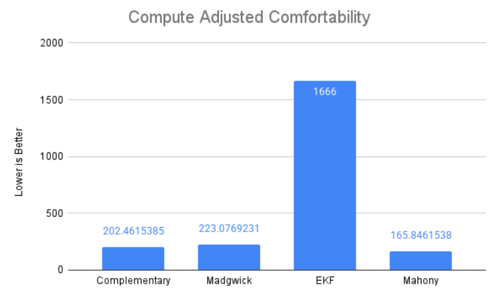

Figure 4. Average comfort times # of operations (lower is better).

3.3. Procedure

To ensure consistent experimental conditions across all participants, we implemented a standardized testing protocol. Each user first completed a 5-minute familiarization period with the VR environment using the linear complementary filter to establish baseline comfort with the system. Subsequently, users attempted to escape the maze for 2.5 minutes using each of the four algorithms in randomized order. After each trial, participants rated their tracking experience on a 10-point scale, focusing on perceived smoothness, responsiveness, and overall comfort. To prevent bias, participants were not informed that the orientation tracking algorithm varied between trials.

4. Results

With data from 16 different users, 14 of them were able to notice orientation algorithm changes.

4.1. Quantitative

Fourteen participants ranked the four algorithms in order of preference on a comfort scale, where lower values indicate better user experience. The Mahony filter achieved the highest user satisfaction with an average ranking of 1.69, followed by Madgwick at 1.92, EKF at 2.62, and complementary filter at 3.62. These results are summarized in Figure 3. The computational requirements varied dramatically across algorithms. The complementary filter ($\alpha = 0.05$) required approximately 56 operations per update, while the Mahony filter ($K_p = 0.9$, $K_I = 0.02$) needed 98 operations. The Madgwick filter ($\beta = 0.05$) consumed 116 operations, and the EKF ($\sigma_Q^2 = 0.001$, $\sigma_R^2 = 0.2$) demanded 637 operations per cycle. To examine the relationship between computational cost and user satisfaction, we computed a ”computational efficiency metric” by multiplying each algorithm's comfort ranking by its operation count. This metric is shown in Figure 4.

4.2. Qualitative

User feedback revealed distinct perceptual differences between algorithms that aligned with their underlying mathematical approaches. The complementary filter was consistently described as ”jumpy” or ”jerky,” reflecting the absence of sophisticated noise filtering in this basic sensor fusion approach. Without integral error correction or gradient-based smoothing, rapid sensor fluctuations translated directly into orientation updates.

The Mahony filter received the most positive qualitative feedback, with users describing the experience as ”very smooth and natural.” This perceived smoothness corresponds to the filter's PI controller architecture, where the integral term accumulates error over time to provide stable, consistent corrections that reduce high-frequency noise and sudden orientation jumps.

Users characterized the Madgwick filter as ”smooth yet delayed,” noting a perceptible lag in response to head movements. This delay reflects the gradient descent optimization process, where smaller step size parameters prioritize stability and noise reduction at the cost of responsiveness. The algorithm's iterative approach to minimizing orientation error inherently introduces latency compared to direct correction methods.

The EKF implementation elicited mixed responses, with users describing it as ”jittery but responsive.” This apparent contradiction stems from the filter's sensitivity to sensor calibration accuracy. While the EKF responded quickly to genuine head movements (responsiveness), the uncalibrated accelerometer introduced measurement uncertainties that propagated through the covariance matrix, manifesting as small but perceptible orientation jitter during stationary periods.

5. Conclusion

The quantitative and qualitative results converge on a clear conclusion: the Mahony filter provides optimal performance for compute-constrained VR environments, contradicting the conventional assumption that available computational resources should be fully utilized. This finding directly challenges the "compute is free" mentality prevalent in modern engineering, demonstrating that algorithmic sophistication does not guarantee superior user experience.

The poor performance of the EKF was particularly surprising given its theoretical advantages and superior performance in controlled studies such as Çoçoli and Badia [4]. I attribute this discrepancy primarily to sensor calibration limitations. The EKF's probabilistic framework relies heavily on accurate noise models and calibrated sensor inputs; without proper accelerometer and gyroscope calibration, the filter's predictions become unreliable, leading to the observed jittery behavior despite consuming 11× more computational resources than simpler alternatives.

Several factors may have influenced the results and warrant consideration in future research. Recency bias could have affected participant ratings, where experiencing a smooth algorithm immediately after a poor one (or vice versa) may have skewed comparative scores. To mitigate this bias in future studies, we recommend implementing randomized algorithm presentation orders with washout periods between trials.

The absence of magnetometer data represents another significant limitation. While our IMU included a magnetometer, it remained uncalibrated and was therefore excluded from all algorithms. Properly calibrated three-axis magnetometer fusion could potentially improve EKF performance by providing additional heading reference information and reducing yaw drift, particularly for the more sophisticated filters.

Parameter optimization across diverse movement patterns proved challenging. VR applications encompass both rapid head movements ($> 200°$/s) and subtle tracking adjustments ($< 10°$/s), making it difficult to find filter parameters that perform optimally across this full dynamic range. Future work should investigate adaptive parameter schemes that adjust filter gains based on detected movement intensity or implement motion-dependent switching between algorithm configurations. Despite these limitations, the results provide compelling evidence that computational efficiency should be prioritized over algorithmic complexity in practical VR implementations, particularly when sensor calibration infrastructure is limited—a common constraint in consumer VR development.

References

[1] S. M. LaValle, A. Yershova, M. Katsev, and M. Antonov. Head tracking for the oculus rift. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[2] X. Li and Y. Wang. Evaluation of ahrs algorithms for foot-mounted inertial-based indoor navigation systems, 2019.

[3] S. O. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 2010.

[4] K. Çoçoli and L. Badia. A comparative analysis of sensor fusion algorithms for miniature imu measurements. *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 239–244, 2023.