HCI Proposal
3D Rotation Mouse

Ignacio María Gallego Torres
Robert Erdmann

January 15, 2026

1 Problem

Viewing virtual 3D objects is especially challenging for users that have no experience with CAD or 3D modeling software. This is because they did not get used to the unintuitive way of controlling 3D objects with conventional 2D input devices yet. The need for this group of users to be able to view virtual 3D objects grows as there usage expands into consumer territories. For example furniture companies use 3D objects to give there customers a better feel for the actual product, when online shopping.

Controlling a 3D scene with a mouse or trackpad is unintuitive. This is due to the two dimensional nature of the conventional input devices.

As three-dimensional (3D) graphics are very important in many computer systems and applications, the search for usable input devices for 3D object manipulation has become a really important topic. In 2D Graphical User Interfaces, the mouse and keyboard became the most common used devices.

However, in 3D interfaces—such as CAD, VR, or complex gaming—there is still no "obvious winner" suitable for all applications.

To effectively manipulate objects in space, a user normally requires six degrees of freedom. The challenge lies in the fact that standard input devices are 2D. Mapping these limited inputs to 3D tasks are often mentally demanding as it requires decomposing into different axes, which can be cognitively difficult. To address this, we were interested in comparing a standard, 2D input method (the keyboard) against a dedicated 3D sensor (a gyroscope).

While various "flying mice" and 3D controllers exist, our goal is to specifically analyze the performance differences in continuous path movement. Unlike static docking tasks where only the final position matters, continuous path manipulation requires the user to follow a dynamic trajectory, maintaining coordination during time. We aim to determine if a 3D input device (gyroscope), which allows for simultaneous rotation across multiple axes, offers a significant advantage over a 2D input device (keyboard) that typically necessitates time-multiplexing - mentally and physically decomposing the movement into separate, sequential steps (e.g., rotating X, then Y, then Z) rather than performing them simultaneously.

1.1 Hypothesis

Our hypothesis is based in the theory of coordination efficiency. We predict that the Gyroscope (3D input) will significantly outperform the Keyboard (2D input) in tasks requiring continuous path rotation.

We hypothesize that the keyboard will force users to separate rotations into individual X, Y, and Z components, and so ending with lower coordination.

The gyroscope allows users to map their hand rotation directly to the object, enabling simultaneous axis manipulation and a smoother movement. Therefore we hypothesize that the gyroscope will show significantly higher efficiency (lower path deviation) than the keyboard, due to the ability to integrate DOFs.

We hypothesize that the keyboard will show a steep learning curve (performance improving over time), whereas the Gyroscope will show flat performance (intuitive from the start)

2 Literature Review

The challenges of 3D input have been extensively categorized by Shumin Zhai, who distinguishes between "free-moving" (isotonic) devices and "desktop" (isometric/elastic) devices. A critical metric in this field is coordination—the ability to simultaneously move multiple degrees of freedom toward a target. Notes that standard 2D mapping techniques often fail to support coordination because users cannot mentally decompose orientation into separate axes effectively.

Isotonic 6 DOF devices (like the "flying mouse" or our Gyroscope) typically outperform other devices in speed, particularly for novice users, because they allow users to map hand motion directly to object motion Other researchers conducted a comparative study of mouse, tactile, and tangible inputs for 3D manipulation.

They found that tangible (3D) interaction was significantly faster than mouse interaction—often by a factor of two. This speed advantage is attributed to the "integrated" nature of 3D input, which allows users to control multiple degrees of freedom simultaneously, whereas mouse/keyboard users are forced to separate interaction modes.

Contrary to the assumption that mouse input is inherently more precise, their results showed that mouse, tactile, and tangible inputs achieved similar levels of accuracy for docking tasks. This suggests that while 3D input is faster, it does not necessarily sacrifice precision.

They observed significant learning effects for the mouse condition, where performance improved notably after the first set of trials. In contrast, the tangible (3D) condition showed no evidence of learning effects, suggesting it is immediately intuitive and requires little training.

While most studies focused on a docking task (reaching a static final position), our research extends this to continuous path movement.

3 Approach

To evaluate the efficacy of 2D vs. 3D input for continuous rotation, we designed a within-subjects experiment comparing two specific conditions:

2D Input (Keyboard): Users control rotation using keys. As noted by some researchers, this modality forces DOF separation, requiring users to switch between axes sequentially.

3D Input (Gyroscope): Users control rotation by moving a handheld device. The device's physical orientation maps directly to the virtual object. This enables DOF integration, allowing simultaneous rotation across multiple axes.

Continuous Path Following will be presented with a 3D object that rotates along a pre-defined, complex trajectory. The user must manipulate their input device to keep their controllable object aligned with the target object's orientation in real-time.

3.1 Prototype / Embedded Gyroscope

To evaluate the comparative performance of 2D (keyboard) versus 3D (gyroscope) input modalities, we developed a software system.

The architecture separates hardware data acquisition from the experimental visualization. The PC acts as the server, running a multi-threaded Python application that renders the 3D environment, processes input, and logs performance metrics in real-time.

The RP2040 microcontroller acts as the client, acquiring raw sensor data and broadcasting it via UDP.

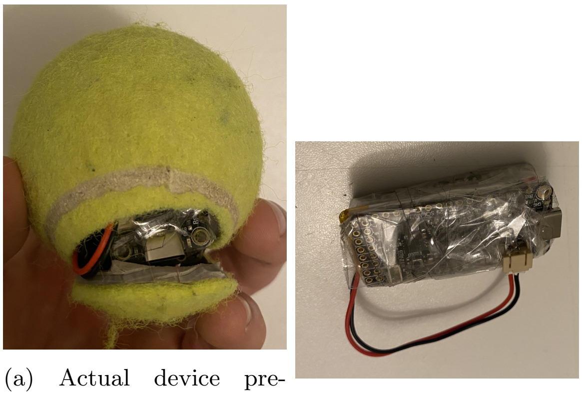
The 3D input device was prototyped using a Raspberry Pi Pico W (RP2040). The firmware, written in MicroPython establishes a Wi-Fi connection to the host server. The system reads the gyroscopic rate of change and orientation.

3.2 Experimental Application

The core experiment was developed using the Ursina Engine. The application manages the experimental logic through a finite state machine.

Unlike static docking tasks, our experiment requires continuous path tracing. We implemented a curve generation algorithm that constructs a 3D trajectory using sinusoidal functions.

Crucially, the system assigns a specific rotation to every segment of the path. This creates a dynamic 3-DOF requirement where the user must continuously adjust Pitch (x), Yaw (y), and Roll (z) to match the cable's orientation.



(a) Actual device presented to participants (b) Embeded Gyroscope

To enforce precision, we implemented another algorithm. The forward velocity of the player is not controlled manually but is a function of the alignment Score. The Alignment Score is basically the angular difference between the player's rotation and the target rotation.

The system includes a data recorder class that logs at every frame. Data is timestamped and exported to CSV files.

The application was designed so that it could be use with 2 modes:

Keyboard (Rate Control): Key presses (W/S, A/D, Q/E) map to rotational velocity. This forces the user to decompose complex rotations into separate axis adjustments, simulating standard 2D interface constraints.

Gyroscope (Position Control): Rotational data from the UDP receiver is mapped directly to the player avatar's orientation. This allows for integrated, simultaneous adjustment of all three axes.

To mitigate learning effects, a dedicated training module was implemented (training_task.py). This module breaks down the 3-DOF task into seven progressive stages. It utilizes an Axis-Locking Mechanism, where inactive axes are interpolated to zero. This allows participants to practice specific rotations (e.g., "Z-axis only") before attempting the fully integrated 3-axis task.

For further information you can check the GitHub Repository

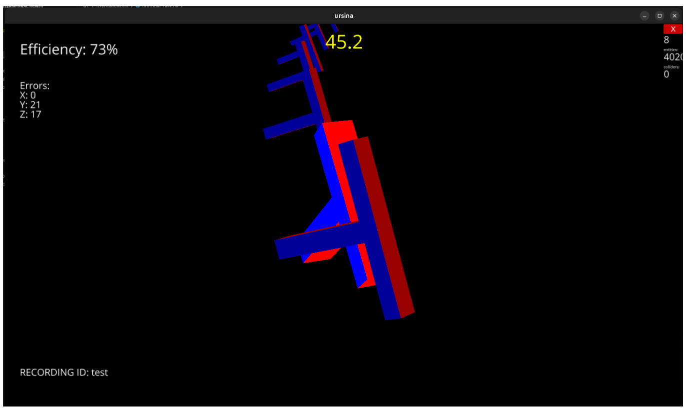


Figure 2: Test 2 Axis with keyboard

4 Analysis

This section evaluates the experimental data to determine how 2D and 3D input modalities differ in continuous rotation tasks. The analysis focuses on the trade-off between *coordination efficiency* (simultaneous axis movement) and *control stability* (precision and overshoot).

4.1 Subjective Workload (NASA-TLX)

The NASA-TLX results (Figure 3) highlight a distinct gap in user comfort and perceived demand. Participants reported significantly higher *Temporal Demand* and lower *Perceived Performance* when using the gyroscope.

Crucially, the "Experience" metric indicates a strong familiarity bias; participants rated themselves as highly experienced with keyboards but novices with 3D spatial controllers. This lack of familiarity likely contributes to the higher perceived effort for the gyroscope, despite its theoretical advantage in 3D tasks.

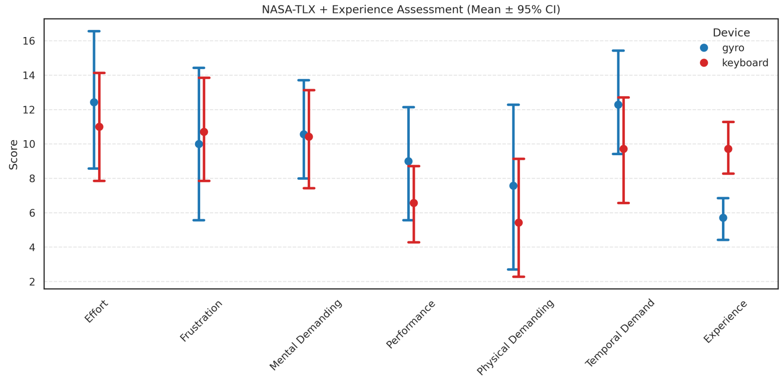


Figure 3: NASA TLX Results showing higher demand for the Gyroscope.

4.2 Performance and Complexity Costs

Objective performance was measured by the total distance traveled along the path (Score). Figure 4 reveals a critical interaction between input device and task complexity.

The Keyboard's "Complexity Cost": The keyboard outperformed the gyroscope in the 2-Axis test. However, its performance dropped significantly when the third axis was introduced. This suggests that the keyboard is efficient for lower-dimensional tasks but suffers a high cognitive or physical penalty when scaling to 3 degrees of freedom (DOF).

The Gyroscope's Stability: Conversely, the gyroscope showed a much smaller performance delta between the 2-Axis and 3-Axis tests. While its average performance was lower than the 2-Axis keyboard, its consistency across complexity levels supports the hypothesis that 3D inputs handle higher-dimensional spaces more naturally, without the steep performance penalty observed in 2D devices.

4.3 Control Strategy: Time-Multiplexing

The most significant finding regarding control strategy is evident in Figure 7. This graph visualizes the percentage of time participants spent moving one, two, or three axes simultaneously.

Keyboard (Sequential): Keyboard users almost exclusively utilized *Time-Multiplexing*. They decomposed the rotation into sequential steps, rotating primarily one axis at a time. This confirms that 2D input devices force users to mentally separate DOFs, creating a bottleneck in complex 3D tasks.

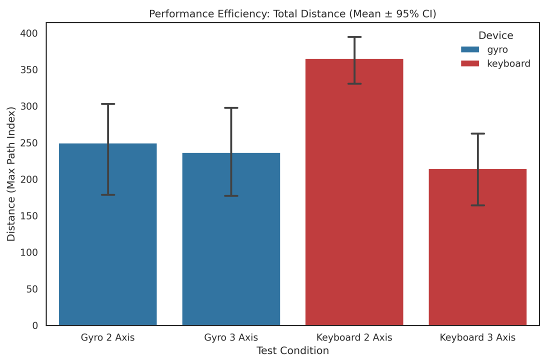


Figure 4: Test Score: Note the sharp drop for Keyboard in 3-Axis tasks.

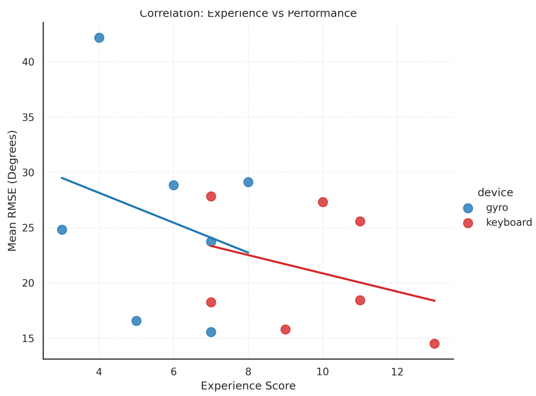


Figure 5: Correlation between self-assessed experience and error rate.

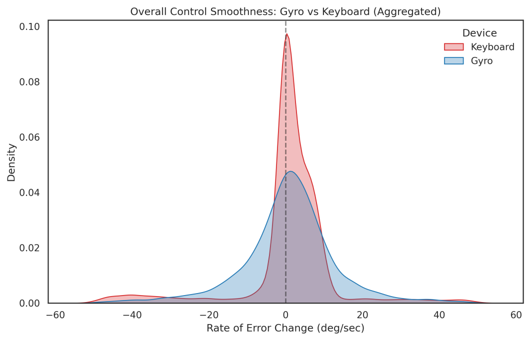


Figure 6: Movement Velocity Density.

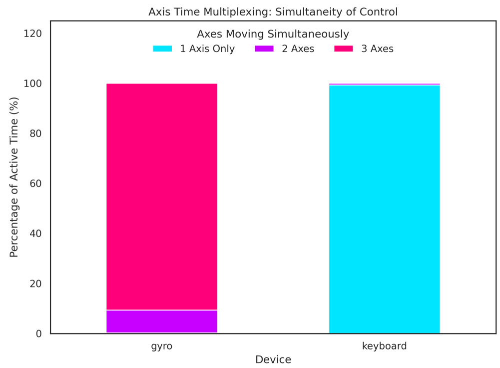


Figure 7: Simultaneous Axis Movement: Keyboard relies on single-axis inputs.

Gyroscope (Simultaneous): In stark contrast, gyroscope users spent the majority of the time manipulating all three axes simultaneously. This confirms the device's capability for DOF integration. However, the fact that this superior coordination did not translate into a winning score (Figure 4) suggests that while the *strategy* was efficient, the *execution* was hampered by other factors, such as precision control.

4.4 Control Quality and Error Distribution

To understand why the gyroscope's superior coordination did not result in higher scores, we analyzed error distribution and velocity.

Variance and Overshoot: The error distribution (Figure 8) and velocity density (Figure 6) show that the gyroscope performance is significantly more "spread out." The velocity plot indicates that gyroscope users frequently moved with high velocity towards the target (high positive values) but also exhibited significant movement away from the target (negative values/overshoot). This points to a lack of damping or stability; users could reach the target rotation quickly but struggled to maintain it without overshooting.

Axis Specificity: The error distribution also highlights that the Z-axis (Roll) was consistently the most difficult axis for users to control, particularly with the gyroscope. This is likely due to anatomical constraints of the human wrist, which has a limited range of motion for roll compared to pitch and yaw.

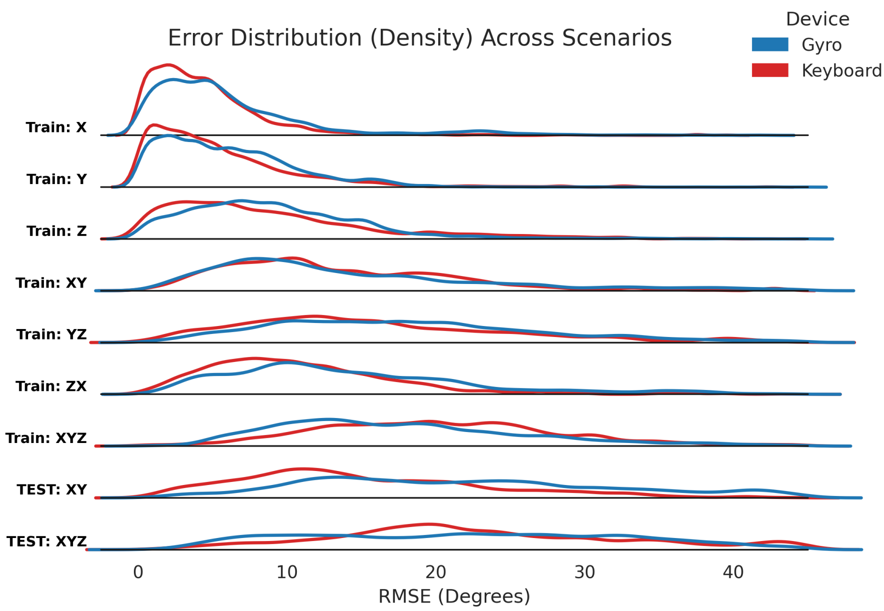


Figure 8: Ridgeline plot showing wider error spread for Gyroscope and Z-axis difficulty.

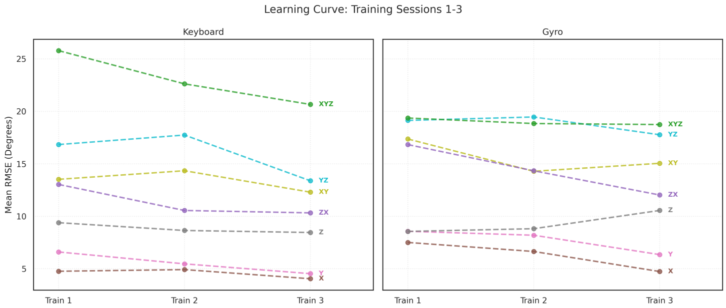


Figure 9: Learning Curve across 3 training sessions.

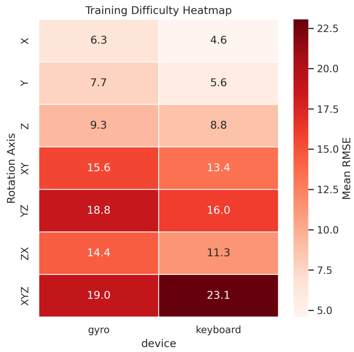


Figure 10: Training Error Heatmap.

4.5 Learning Curve

The learning curve (Figure 9) demonstrates a clear progression across the three training sessions. The downward trend in error rates confirms that the task required skill acquisition. Consistent with the error distribution analysis, the heatmap (Figure 10) confirms that the Z-axis and combined YZ-axis rotations remained the primary friction points throughout the training phase.

4.6 Summary of Analysis

The data reveals a distinct trade-off between the two input modalities.

The **Keyboard** facilitates stable, precise control through rate-limiting, but it forces users into a cognitive bottleneck of time-multiplexing. This strategy is highly effective for simple (2-Axis) tasks but scales poorly, resulting in significant performance degradation when a third axis is added.

The **Gyroscope** enables true 3-DOF interaction, allowing users to manipulate three axes simultaneously without mental decomposition. However, this freedom comes at the cost of stability. The performance data is characterized by high variance and overshoot, indicating that while users *can* rotate freely, they struggle to stop precisely.

Ultimately, the gyroscope allows for the correct *type* of movement (simultaneous 3D rotation), but the lack of physical constraints (like the friction of a mouse or keys) leads to involuntary movements and tremors that hamper the final score.

5 Conclusion

Contrary to our initial hypothesis regarding the superiority of 3D input, the results indicate a really high advantage for the Keyboard (2D input) during the 2-Axis rotation tasks. Participants demonstrated greater precision when restricting movement to only two dimensions (Pitch and Yaw), suggesting that the cognitive load of decomposing 2D rotation is manageable with a standard input device.

In the more complex 3-Axis condition (Pitch, Yaw, and Roll), the performance gap between the two modalities largely disappeared. The results were comparable, indicating that while the keyboard loses its advantage as complexity increases, the gyroscope did not provide the expected dominant improvement in overall completion time for this specific implementation.

As hypothesized, the Gyroscope excelled in simultaneous axis manipulation. Data logs confirm that gyroscope users were able to adjust Pitch, Yaw, and Roll concurrently, achieving a higher degree of integrated movement compared to the step-by-step inputs observed in keyboard users.

Despite better coordination, the gyroscope suffered from volatility in speed control. The involuntary hand tremors or over-rotations translated directly into erratic movements causing users to fluctuate between moving too fast or too slow. In contrast, the keyboard provided a stable rate control mechanism, allowing users to maintain a consistent velocity, which contributed to its superior performance in lower-complexity tasks.

6 Further Work

For future work, we plan to expand the scope of comparison by including additional devices. Participant feedback suggested that the shape of the 3D input device should more closely mimic the geometry of the corresponding virtual object. Additionally, utilizing a higher-precision gyroscope is recommended; the current sensor required frequent recalibration, which may have negatively influenced the NASA TLX scores by increasing the users' perceived physical and mental demand.

Furthermore, a larger dataset would have been beneficial to analyze how demographic features, such as age or prior experience, influence performance. Finally, software optimization is required to address latency issues observed during testing

7 References

(2023). Imag.fr. http://iihm.imag.fr/coutrix/ens/M2-MoSIG-ARNIT/projects/2023-2024/JonAle.zip

Zhai, S. (1998, november). User performance in relation to 3D input device design. ACM SIGGRAPH Computer Graphics, 32(4), 50-54. https://doi.org/10.1145/307710.307728

Besançon, Lonni, et al. "Mouse, tactile, and tangible input for 3D manipulation." Proceedings of the 2017 CHI conference on human factors in computing systems. 2017. https://inria.hal.science/hal-01436206/file/chi2017.pdf