Creation of a Pre-Recorded Motion Library

Technical Report

Abstract

We developed a reusable library of naturalistic cursor-motion snippets to replace synthetic Ornstein–Uhlenbeck (OU) noise in our Control–Detection task. Participants each provided 5 minutes of continuous mouse movements under task-like instructions. We segmented these traces into 5 s (300-frame) snippets, standardized them, and stored them as a master pool. This report details participant recruitment, recording apparatus, initial OU-based demo, data preprocessing, snippet generation, and library assembly.

1 Introduction

Psychophysical tasks that probe agency rely on mixing participants' real movements with "noise" to create stimuli of varied controllability. While OU noise offers parametric control, it lacks the complex kinematic structure of human motion. Here, we capture and curate real cursor trajectories from multiple participants to form a *motion library*—a database of discrete, five-second motion patterns reflecting naturalistic hand dynamics.

2 Methods

2.1 Apparatus & Software

- **Hardware:** Desktop PCs with 60 Hz monitors and standard optical mice.
- **Software:** PsychoPy v2025.2, Python 3.10, NumPy v1.24.

2.2 General Procedure

Participants were instructed to move their mouse cursor for 5 min trying to notice which of two shapes responded more to their movements. After every 10 seconds the two shapes froze and their task was to identify which one they thought they could control more by clicking on them. The correct shape turned green while the incorrect shape turned red. In order to keep engagement and hence incentive to show continous movements high, participants could collect points for correct responds displayed on the top left corner of the screen - outside of the are the shapes could move towards. To increase the gamification aspect further, we also added a timer counting down from 5min.

2.3 Recording Procedure

Participants ran a full-screen PsychoPy script for 5 minutes:

1. *Instruction Screen:* "Press `S` to START; `ESC` to QUIT."
2. *Main Loop (300 s @60 Hz = 18 000 frames):*
   - Record raw mouse position (x_t, y_t) each frame.
   - Compute deltas
     $$\Delta x_t = x_t - x_{t-1}, \quad \Delta y_t = y_t - y_{t-1}.$$
   - Blend controlled shape with Δ and distractor with noise (initially OU in the demo).
   - Display two shapes confined to radius 250 px.
   - Every 10 s, pause; participant clicks the shape they believe they controlled (correct $\to$ green, incorrect $\to$ red); update score.
3. *Data Buffer:* Store
   $$\text{trace} \in R^{18000\times 2},$$
   where each row is $(\Delta x_t, \Delta y_t)$.
4. *Save:* Write `trace` to disk on completion or quit.

2.4 OU-Based Demo

During the entire 5 min a synthetic Ornstein–Uhlenbeck (OU) noise for the unattended shape was used. The OU process is a mean-reverting stochastic differential equation:

$$d\mathbf{v}_t = -\theta\,\mathbf{v}_t\,dt + \sigma\,d\mathbf{W}_t,$$

where $\mathbf{W}_t$ is a 2D Wiener process, θ governs the strength of reversion to zero, and σ scales the noise. Discretized at $dt = 1/60$ s:

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \theta\,\mathbf{v}_t\,dt + \sigma\sqrt{dt}\,\boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Parameters used:

$$\theta = 0.2, \quad \sigma = 0.8, \quad dt = \tfrac{1}{60}, \quad T_{\text{demo}} = 30\,\text{s} \mathrel{\widehat{=}} 1800 \text{ frames}.$$

At each frame t, the controlled shape moves by the actual mouse delta $(\Delta x_t, \Delta y_t)$, while the distractor moves by $\mathbf{v}_t$ from the OU snippet. After 30 s, we estimate linear bias

$$\bar{\Delta} = \frac{1}{1800}\sum_{t=1}^{1800}(\Delta x_t, \Delta y_t),$$

and angular bias

$$\overline{\Delta\theta} = \frac{1}{1799}\sum_{t=1}^{1799}\big[\arg(\Delta x_{t+1}, \Delta y_{t+1}) - \arg(\Delta x_t, \Delta y_t)\big].$$

2.5 Snippet Segmentation & Standardization

After recording, each participant's trace is processed:

1. **Usable Frames:**
$$\text{usable} = \left\lfloor \tfrac{18000}{300} \right\rfloor \times 300 = 18000.$$

2. **Reshaping:**
$$\text{snips} = \text{trace}[:\text{usable}]\ \text{reshape}(-1, 300, 2),$$
yielding 60 snippets per participant.

3. **Z-scoring:** For each snippet $s \in R^{300\times 2}$, compute
$$\mu_s = \frac{1}{300}\sum_{t=1}^{300} s_t, \quad \sigma_s = \sqrt{\frac{1}{300}\sum_{t=1}^{300} \|s_t - \mu_s\|^2},$$
then standardize
$$s'_t = \frac{s_t - \mu_s}{\sigma_s + 10^{-9}}.$$

4. **Storage:** Convert to `float16` and save each participant's array as `participant_<ID>_<timestamp>.`

2.6 Master Library Assembly

Concatenate all participants with:

```
import numpy as np, pathlib
files = sorted(pathlib.Path("motion_participants").glob("*.npy"))
all_snips = np.concatenate([np.load(f) for f in files], axis=0)
np.save("master_snippets.npy", all_snips.astype(np.float16))
```

If each participant yields 60 snippets at a samplesize of 10 participants, the resulting array has shape $(600, 300, 2)$.

3 Results

- *Participants recorded:* 10
- *Total snippets:* 600
- *Snippet length:* 300 frames $\approx 5\,\text{s}$
- *Master library size:* $\sim$42 MB

4 Discussion

Using real cursor kinematics preserves natural variability—speed bursts, hesitations—not captured by parametric noise. Five-second snippets balance snippet independence with temporal context. Z-scoring ensures amplitude uniformity across participants.

5 Conclusion

We present a complete workflow—from participant recording to library assembly—for a high-fidelity motion library, enhancing the realism of our control-detection experiment.