The Journal of Open Source Software

visual-gestures.js- Lets you control the cursor with hand or finger movements

Nagendra Dharmireddi [1*], **Boddu Sri Pavan** [1*], **Boilla Lavanya** [1*], and **Boddu Swathi Sree** [1*]

1 Rajiv Gandhi University of Knowledge Technologies, Nuzvid * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Sébastien Boisgérault

Reviewers:

- @chand-lab
- @boisgera

Submitted: 15 February 2025
Published: unpublished

Summary

visual-gestures.js is an open-source TypeScript library enabling precise cursor control hover, click, drag, and drop through hand gestures in the air. It replaces traditional touch-based interactions with intuitive, natural gestures, offering seamless performance, offline support, and uninterrupted productivity. Designed with an event-driven architecture, it is highly extensible across diverse applications. The library is open-source and available on GitHub : https://github.com/learn-hunger/visual-gestures

Statement of need

Traditional cursor control relies on touch-based interaction, requiring physical contact with a device. Transitioning to gesture-based control enhances user experience across various domains, including AR, VR, gaming, healthcare, e-commerce, and industrial automation.

visual-gestures.js addresses the need for a ready-to-integrate, open-source solution, offering seamless integration, offline functionality, and high customizability, including debugger support for rapid development.

Proposed Methodology

We introduce pseudo events, gesture-based equivalents of traditional cursor events, enabling seamless human-computer interaction without physical touch. Gesture recognition is achieved by extracting hand landmarks using Google MediaPipe's Hand Landmarker.

At the core of our approach is the Fluid Kink Ratio Algorithm (FKR Algorithm), a novel technique, we propose, for gesture-based decision-making. FKR Algorithm dynamically tracks the ratio of two key segment lengths: (i) Index fingertip to its base and (ii) Index finger base to wrist base. This ratio is monitored within a hibernating sliding window of the last five recorded frames, ensuring smooth tracking and reducing transient noise.

Instead of conventional Euclidean distance, we utilize Weighted Euclidean distance, which assigns greater importance to the y-axis component, enhancing vertical gesture sensitivity.

$$\text{Weighted Euclidean Distance} = \sqrt{\alpha(x_1 - x_2)^2 + \beta(y_1 - y_2)^2}$$
$$where\ \alpha,\ \beta \in [0, 1]$$

Figure 1, 2 illustrate the sequential gesture transitions within the sliding window, visualized using KitikiPlot, depicting movement trends across consecutive time frames.

The Journal of Open Source Software

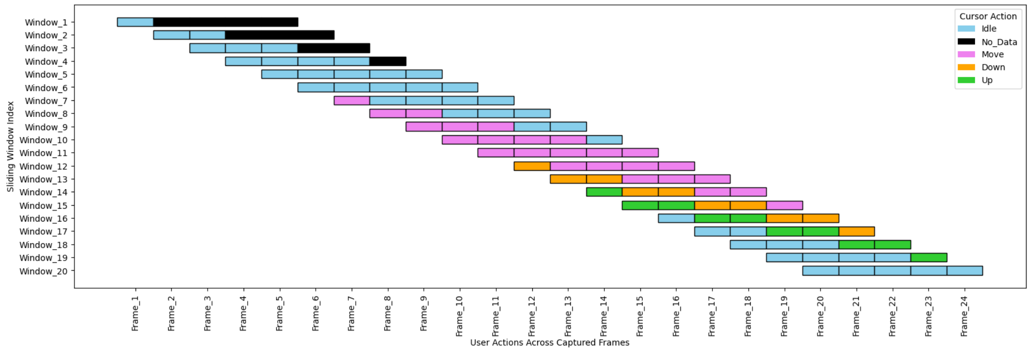

Figure 1: Visualization of user action sequences using KitikiPlot where the x-axis represents user action detected for each frame, and the y-axis consecutive sliding windows of length 5, stride 1, where distinct colors represent different events, enabling clear identification and differentiation of user interactions during gesture-based operations.

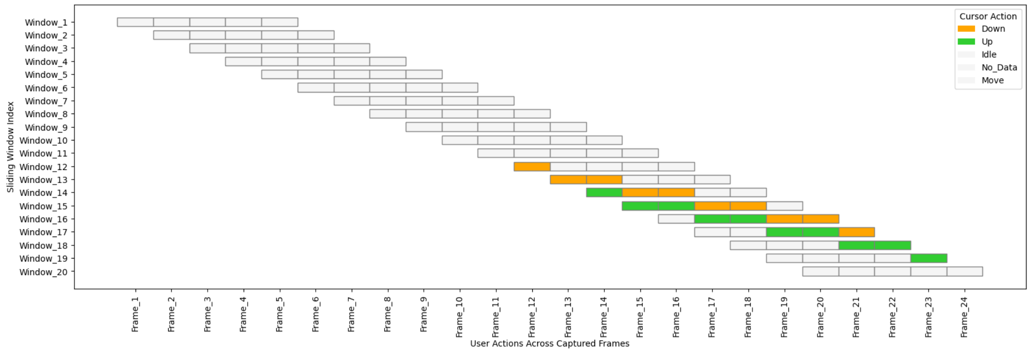

Figure 2: Sliding window visualization focusing specifically on cursor up and down events, where the x-axis represents user action detected for each frame, and the y-axis consecutive sliding windows of length 5, highlighting the frame-level sequential relationship within the user gesture actions.

We developed a robust debugging panel, as shown in Figure 3 featuring memory and CPU utilization tracking, framerate monitoring, landmark skeletal visualization, a live finger kink ratio graph, and a customizable debug UI. This enables developers to extend and tailor the system to their specific use cases efficiently.

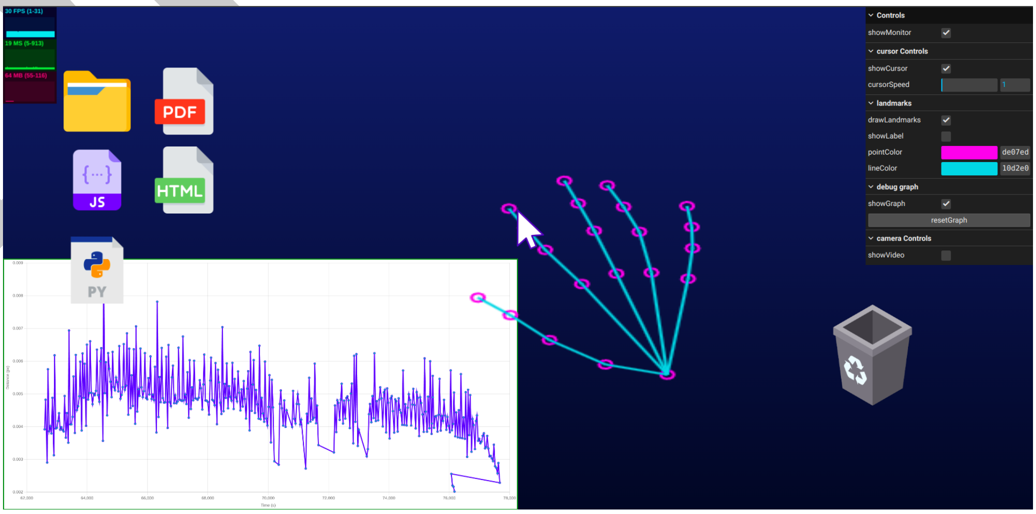

Figure 3: Comprehensive Debugging Panel

Acknowledgements

We express our heartfelt gratitude to our parents and family members for their unwavering support. This work is conducted with the gracious blessings of Paramatma.

References

google-ai-edge. (2019). mediapipe: Cross-platform, customizable ML solutions for live and streaming media. *GitHub*. https://github.com/google-ai-edge/mediapipe

google-ai-edge. (2019). MediaPipe Hand Landmarker (Hand Landmarks Detection Guide). *Google AI Edge*. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker

Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, Cristian Sminchisescu. (2020). GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6184-6193. https://openaccess.thecvf.com/content_CVPR_2020/html/Xu_GHUM__GHUML_Generative_3D_Human_Shape_and_Articulated_Pose_CVPR_2020_paper.html

Boddu Sri Pavan, Chandrasheker Thummanagoti, & Boddu Swathi Sree. (2025). KitikiPlot A Python library to visualize categorical sliding window data. *Zenodo*. https://doi.org/10.5281/zenodo.14632005

Nagendra Dharmireddi, & Jhansi Pyla. (2024). visualise-data-kit: Skeleton based visualizations of Google's MediaPipe. *GitHub*. https://github.com/learn-hunger/visualise-data-kit