

1. Inspiration papers:

Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S. (2010). De-signing gaze gestures for gaming: An investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications*, ETRA '10, page 323–330, New York, NY, USA. Association for Computing Machinery.

Drewes, H. and Schmidt, A. (2007). Interacting with the computer using gaze gestures. In *Human-Computer Interaction–INTERACT 2007: 11th IFIP TC 13 International Conference, Rio de Janeiro, Brazil, September 10-14, 2007, Proceedings, Part II* 11 pages 475–488. Springer Berlin Heidelberg.

Both inspired a gaze gesture model

Drewes & Schmidt's saccade direction method more appealing for VR

2: SRAnipal

HTC's SDK for animating player avatars in VR (Super Reality Animation pal)

Plugins for game engines allow them to access gaze data

GetGazeData provides normalized gaze vector

No inherent fixation/saccade recognition or blink handling, must be implemented

- Hacky blink implementation; if a blink is in progress, gaze is assumed to be the same as it was on the last frame before the blink, which still sometimes causes unintentional “saccades”
- Better implementation may ignore comparisons during blink and a few frames after

3. Fixations & Saccades

Identify saccades by looking for rapid eye movement between frames

If the angle between gaze vectors crosses a threshold a saccade is in progress

- Threshold is (.15°/frame, 13.5°/s), large enough to be safely not a fixation but slow enough to capture even slow saccades
- Angle between vectors calculated via $\arccos(a \cdot b / (|a||b|))$, no projecting points onto planes

If a saccade is long enough, it can be part of a gesture

During no saccade, assume fixation & keep track of duration

Drop prior saccades if fixation is adequately long

4. Direction Classification

Remember gaze vector from beginning of saccade

Compare to vector at end of saccade

High +/- difference in X indicates up/down, in Y indicates right/left

- Brief halts or system errors in the middle of a saccade can cause it to be recognized as multiple saccades in the same direction, interfering with gesture input
- Multiple consecutive saccades in the same direction ignored, should not be used as gesture input

Diagonals ignored, both for system simplicity and to maintain eye tracking accuracy

- See notes on UI grid

5. Head-Based Selection

Use camera's `GetForwardVector` to cast a ray matching HMD orientation

Raycast returns hit objects; process gaze only when hitting an interactable

UI object attached to head (rendered in a canvas space) helps guide saccades

- 3x3 grid of 9 points

Allows for many gestures, but eye tracking accuracy is

compromised when looking at a corner point; losing tracking briefly

causes the system to “recognize saccades” in error

- Grid of 4 points; one centered and one point for each cardinal direction

Allows for fewer gestures, but maintains eye tracking accuracy at all points

Interactable objects have a `recieveGesture` function

- When an object is selected, its `recieveGesture` function is called every frame to check if it should perform an action based on the current gesture