Generating Virtual Reality Stroke Gesture Data from Out-of-Distribution Desktop Stroke Gesture Data *Appendices*

Lin-Ping Yuan, Boyu Li, Jindong Wang, Huamin Qu, Wei Zeng

1 Feature Explanation

As stated in Sec. 3.1, we extract geometric and kinematic features related to the (x, y) plane at the point level and stroke level for each point in a desktop or VR stroke. This section gives a detailed description and calculation for the full feature list. We represent a stroke as $s = [p_1, p_2, \cdots, p_n]$, where $P_i = [x_i, y_i, t_i]$. For desktop, the stroke is the original sequence of points as captured by the input device. For VR, the stroke is obtained by projecting the original stroke onto the (x, y) plane.

1.1 Point Level Features

1.1.1 *Geometric Features*

Length (Len_i) is the Euclidean distance between the current point p_i and its previous point p_{i-1}.

$$Len_i = \begin{cases} \|p_i - p_{i-1}\| & i \in [2, n] \\ 0 & i = 1 \end{cases}$$

Turning angle (θ_i) measures the sharpness of the stroke at the current point p_i. A larger turning angle θ_i indicates a sharper change in the direction of the stroke at point p_i. Given three consecutive points p_{i-1}, p_i, and p_{i+1}, the turning angle θ_i is the angle between the two vectors $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i p_{i+1}}$. We calculate θ_i as follows:

$$\theta_i = \begin{cases} \arctan \frac{\Delta x_i \Delta y_{i-1} - \Delta x_{i-1} \Delta y_i}{\Delta x_i \Delta x_{i-1} + \Delta y_i \Delta y_{i-1}} & i \in [2, n-1] \\ 0 & i = 1, n \end{cases}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta x_{i-1} = x_i - x_{i-1}$, $\Delta y_i = y_{i+1} - y_i$, and $\Delta y_{i-1} = y_i - y_{i-1}$.

Curvature $(curv_i)$ represents the degree to which the stroke changes direction over a small segment near the current point p_i. It is related to the radius of the circle that best approximates the stroke at p_i. The curvature is larger when the stroke is more sharply curved and smaller when the stroke is more straight. Given three consecutive points p_{i-1}, p_i, and p_{i+1}, the curvature at p_i can be calculated as:

$$curv_i = \begin{cases} \frac{4Area}{\|p_i - p_{i-1}\| \|p_{i+1} - p_i\| \|p_{i+1} - p_{i-1}\|} & i \in [2, n-1] \\ 0 & i = 1, n \end{cases}$$

where $Area$ is the area of the triangle formed by the three points p_{i-1}, p_i, and p_{i+1}, and $\|p_i - p_{i-1}\|$, $\|p_{i+1} - p_i\|$, and $\|p_{i+1} - p_{i-1}\|$ are the lengths of the sides of the triangle.

1.1.2 *Kinematic Features*

Velocity (vel_i) is the position velocity at the current point p_i. Since our goal is to estimate the unknown depth (i.e., z values) given (x, y) values, we only consider the position changes along x and y axis. Thus, velocity v is calculated as:

$$vel_i = \begin{cases} \frac{Len_{i+1} + Len_i}{t_{i+1} - t_{i-1}} & i \in [2, n-1] \\ 0 & i = 1, n \end{cases}$$

Acceleration (acc_i) is calculated as:

$$acc_i = \begin{cases} \frac{vel_i - vel_{i-1}}{t_i - t_{i-1}} & i \in [2, n] \\ 0 & i = 1 \end{cases}$$

Jerk $(jerk_i)$ is calculated as:

$$jerk_i = \begin{cases} \frac{acc_i - acc_{i-1}}{t_i - t_{i-1}} & i \in [2, n] \\ 0 & i = 1 \end{cases}$$

1.2 Segment Level Features

1.2.1 *Geometric features*

Path length $(PathLen_i)$ is the cumulative sum of Euclidean distance between any two adjacent points among the starting point p_1 and the current point p_i.

$$PathLen_i = \sum_{j=2}^{i} Len_j = \sum_{j=2}^{i} \|p_j - p_{j-1}\|$$

Starting and ending point distance $(StartEndDis_i)$ is the straight line Euclidean distance between the starting point p_1 and the current point p_i.

$$StartEndDis_i = \|p_i - p_1\|$$

Line similarity $(LineSim_i)$ measures how closely the stroke follows a straight line path.

$$LineSim_i = \frac{StartEndDis_i}{PathLen_i}$$

Area of the a segment's bounding box $(AreaBBox_i)$ is calculated as follows:

$$AreaBBox_i = (x_{max} - x_{min}) * (y_{max} - y_{min})$$

where $x_{max} = \max(x_1, x_2, \ldots, x_i)$, $x_{min} = \min(x_1, x_2, \ldots, x_i)$, $y_{max} = \max(y_1, y_2, \ldots, y_i)$, and $y_{min} = \min(y_1, y_2, \ldots, y_i)$.

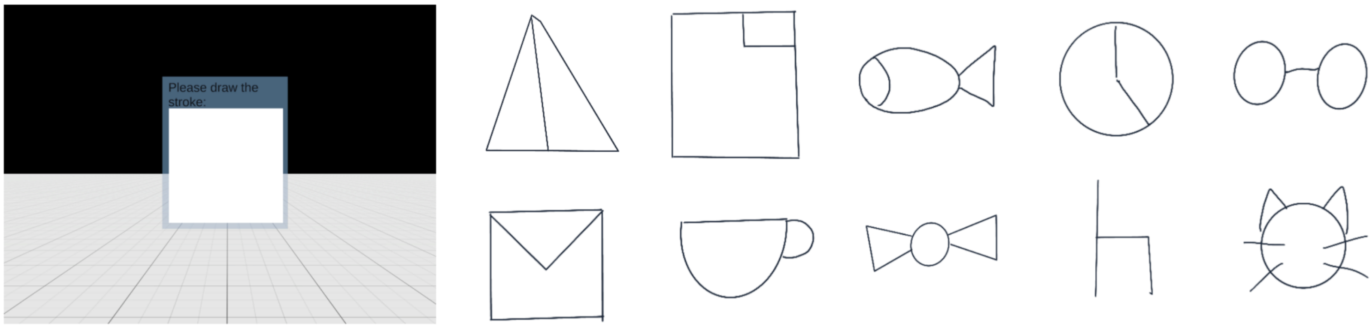

Fig. 1. User interface of our VR drawing program and the ten types of sketches users need to draw.

Length of a segment's bounding box diagonal ($LenBBox_i$) is calculated as follows:

$$LenBBox_i = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}$$

Angle of a segment's bounding box diagonal ($AngelBBox_i$) reflects the global orientation of the segment and is calculated as follows:

$$AngleBBox_i = \arctan \frac{y_{max} - y_{min}}{x_{max} - x_{min}}$$

Total turning angle (Θ_i) is the sum of absolute the turning angle at each point p_i.

$$\Theta_i = \sum_{j=1}^{i} |\theta_j|$$

Overall sharpness ($OverallSharp_i$) is calculated as the squared turning angle at each point p_i.

$$OverallSharp_i = \sum_{j=1}^{i} \theta_j^2$$

Overall curvature ($OverallCurve_i$) is calculated as:

$$OverallCurve_i = \frac{\Theta_i}{PathLen_i}$$

2 Data Collection for VR Stroke Prediction

As stated in Sec. 7.2, we develop a VR drawing program with logging functions to collect multiple VR stroke datasets for VR stroke prediction. Figure 1 shows the user interface and the ten types of sketches we ask users to draw.