Step 1. Resample a *points* path into *n* evenly spaced points. We use *n*=64. For gestures serving as templates, Steps 1-3 should be carried out once on the raw input points. For candidates, Steps 1-4 should be used just after the candidate is articulated.

RESAMPLE(*points*, *n*)
1 $I \leftarrow$ PATH-LENGTH($points$) / (n – 1)
2 $D \leftarrow 0$
3 $newPoints \leftarrow points_0$
4 **foreach** point p_i for $i \geq 1$ in *points* **do**
5 $\quad d \leftarrow$ DISTANCE($p_{i\text{-}1}$, p_i)
6 $\quad$ **if** $(D + d) \geq I$ **then**
7 $\quad\quad q_x \leftarrow p_{i\text{-}1_x} + ((I - D) / d) \times (p_{i_x} - p_{i\text{-}1_x})$
8 $\quad\quad q_y \leftarrow p_{i\text{-}1_y} + ((I - D) / d) \times (p_{i_y} - p_{i\text{-}1_y})$
9 $\quad\quad$ APPEND(*newPoints*, *q*)
10 $\quad\quad$ INSERT(*points*, *i*, *q*) // *q* will be the next p_i
11 $\quad\quad D \leftarrow 0$
12 $\quad$ **else** $D \leftarrow D + d$
13 **return** *newPoints*

PATH-LENGTH(*A*)
1 $d \leftarrow 0$
2 **for** *i* **from** 1 **to** $|A|$ **step** 1 **do**
3 $\quad d \leftarrow d$ + DISTANCE($A_{i\text{-}1}$, A_i)
4 **return** *d*

Step 2. Find and save the indicative angle ω from the *points*' centroid to first point. Then rotate by $-\omega$ to set this angle to 0°.

INDICATIVE-ANGLE(*points*)
1 $c \leftarrow$ CENTROID(*points*) // computes $(\bar{x}, \bar{y})$
2 **return** ATAN($c_y - points_{0_y}$, $c_x - points_{0_x}$) // for $-\pi \leq \omega \leq \pi$

ROTATE-BY(*points*, ω)
1 $c \leftarrow$ CENTROID(*points*)
2 **foreach** point *p* **in** *points* **do**
3 $\quad q_x \leftarrow (p_x - c_x)$ COS $\omega - (p_y - c_y)$ SIN $\omega + c_x$
4 $\quad q_y \leftarrow (p_x - c_x)$ SIN $\omega + (p_y - c_y)$ COS $\omega + c_y$
5 $\quad$ APPEND(*newPoints*, *q*)
6 **return** *newPoints*

Step 3. Scale *points* so that the resulting bounding box will be of $size^2$ size. We use *size*=250. Then translate *points* to the origin *k*=(0,0). BOUNDING-BOX returns a rectangle defined by (min_x, min_y), (max_x, max_y).

SCALE-TO(*points*, *size*)
1 $B \leftarrow$ BOUNDING-BOX(*points*)
2 **foreach** point *p* **in** *points* **do**
7 $\quad q_x \leftarrow p_x \times size / B_{width}$
8 $\quad q_y \leftarrow p_y \times size / B_{height}$
9 $\quad$ APPEND(*newPoints*, *q*)
10 **return** *newPoints*

TRANSLATE-TO(*points*, *k*)
1 $c \leftarrow$ CENTROID(*points*)
2 **foreach** point *p* **in** *points* **do**
3 $\quad q_x \leftarrow p_x + k_x - c_x$
4 $\quad q_y \leftarrow p_y + k_y - c_y$
5 $\quad$ APPEND(*newPoints*, *q*)
6 **return** *newPoints*

Step 4. Match *points* against a set of *templates*. The *size* variable on line 7 of RECOGNIZE refers to the *size* passed to SCALE-TO in Step 3. The symbol φ equals ½(-1 + √5). We use θ=±45° and θ_Δ=2° on line 3 of RECOGNIZE. Due to using RESAMPLE, we can assume that *A* and *B* in PATH-DISTANCE contain the same number of points, i.e., $|A|$=$|B|$.

RECOGNIZE(*points*, *templates*)
1 $b \leftarrow +\infty$
2 **foreach** template *T* in *templates* **do**
3 $\quad d \leftarrow$ DISTANCE-AT-BEST-ANGLE($points$, T, $-\theta$, $+\theta$, θ_Δ)
4 $\quad$ **if** $d < b$ **then**
5 $\quad\quad b \leftarrow d$
6 $\quad\quad T' \leftarrow T$
7 $score \leftarrow 1 - b / 0.5\sqrt{size^2 + size^2}$
8 **return** $\langle T', score \rangle$

DISTANCE-AT-BEST-ANGLE(*points*, *T*, θ_a, θ_b, θ_Δ)
1 $x_1 \leftarrow \varphi\theta_a + (1 - \varphi)\theta_b$
2 $f_1 \leftarrow$ DISTANCE-AT-ANGLE($points$, T, x_1)
3 $x_2 \leftarrow (1 - \varphi)\theta_a + \varphi\theta_b$
4 $f_2 \leftarrow$ DISTANCE-AT-ANGLE($points$, T, x_2)
5 **while** $|\theta_b - \theta_a| > \theta_\Delta$ **do**
6 $\quad$ **if** $f_1 < f_2$ **then**
7 $\quad\quad \theta_b \leftarrow x_2$
8 $\quad\quad x_2 \leftarrow x_1$
9 $\quad\quad f_2 \leftarrow f_1$
10 $\quad\quad x_1 \leftarrow \varphi\theta_a + (1 - \varphi)\theta_b$
11 $\quad\quad f_1 \leftarrow$ DISTANCE-AT-ANGLE($points$, T, x_1)
12 $\quad$ **else**
13 $\quad\quad \theta_a \leftarrow x_1$
14 $\quad\quad x_1 \leftarrow x_2$
15 $\quad\quad f_1 \leftarrow f_2$
16 $\quad\quad x_2 \leftarrow (1 - \varphi)\theta_a + \varphi\theta_b$
17 $\quad\quad f_2 \leftarrow$ DISTANCE-AT-ANGLE($points$, T, x_2)
18 **return** MIN(f_1, f_2)

DISTANCE-AT-ANGLE(*points*, *T*, θ)
1 $newPoints \leftarrow$ ROTATE-BY($points$, θ)
2 $d \leftarrow$ PATH-DISTANCE($newPoints$, T_{points})
3 **return** *d*

PATH-DISTANCE(*A*, *B*)
1 $d \leftarrow 0$
2 **for** *i* **from** 0 **to** $|A|$ **step** 1 **do**
3 $\quad d \leftarrow d$ + DISTANCE(A_i, B_i)
4 **return** $d / |A|$

[1] This pseudocode is modified slightly from that which appears in the original ACM UIST 2007 publication by Wobbrock, Wilson and Li to be parallel to the more recent $*N* multistroke recognizer. This algorithm's logic remains unchanged.