$P Point-Cloud Recognizer Extended to Different Point Cardinalities

Martez E. Mott[1], Radu-Daniel Vatavu[2], Shaun K. Kane[3], Jacob O. Wobbrock[1]

[1]The Information School
DUB Group
University of Washington
Seattle, WA 98195 USA
{memott, wobbrock}@uw.edu

[2]MintViz Lab | MANSiD
Research Center
University Stefan cel
Mare of Suceava
Suceava 720229, Romania
vatavu@eed.usv.ro

[3]University of Colorado, Boulder
Boulder, CO 80309 USA
shaun.kane@colorado.edu

In the following pseudocode, we list a new variant of the $P point-cloud gesture recognizer that we adapted for our problem of predicting touch input locations from varied touch behavior. The new $P variant can match an unequal number of touch areas (i.e. ellipses) that make up templates and candidates.

GREEDY-CLOUD-MATCH-UNEQUAL (POINTS *candidate*, POINTS *template*)

```
n ← MAX(candidate.length, template.length)
ε ← .50
step ←floor($n^{1-\varepsilon}$)
min ← ∞
ptsLeft ← n
if candidate.length > template.length then
    for i = 0 to n step step do
        d ← CLOUD-DISTANCE-UNEQUAL(template, candidate, i, ptsLeft)
        min ← MIN(min, d)
else
    for i = 0 to n step step do
        d ← CLOUD-DISTANCE-UNEQUAL(candidate, template, i, ptsLeft)
        min ← MIN (min, d)
return min
```

CLOUD-DISTANCE-UNEQUAL (POINTS *matcher*, POINTS *matchee*, int *start*, int *ptsLeft*)

```
if ptsLeft == 0 then
    return 0
else
    m ← matcher.length
    n ← matchee.length
    matched ← new bool[n]
    sum ← 0
    i ← start
    do
        min ← ∞
        index ← -1
        for each j such that not matched[j] do
            d ← EUCLIDEAN-DISTANCE($matcher_i$, $matchee_j$)
            if d < min then
                min ← d
                index ← j
        matched[index] ← true
        ptsLeft ← ptsLeft - 1
        weight ← 1 – ((i – start + m) MOD m) / m
        sum ← sum + weight × min
        i ← (i + 1) MOD m
    until  i == start
    diff ← n – m
    remaining ← DETERMINE-REMAINING(matched, matchee, diff)
    if diff < m then
        return sum + CLOUD-DISTANCE-UNEQUAL(remaining, matcher, 0, ptsLeft)
    else
        return sum + CLOUD-DISTANCE-UNEQUAL(matcher, remaining, 0, ptsLeft)
```

DETERMINE-REMAINING (bool[ ] *matched*, POINTS *points*, int *diff*)

```
remaining ← new POINTS[diff]
index ← 0
for each i such that not matched[i] do
    remaining[index] ← points[i]
    index ← index + 1
return remaining
```