

3D Gesture Recognition

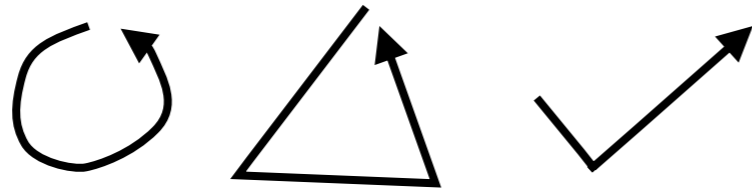
Ben Alun-Jones

Since last time...

- Moved from the N900 to iPhone and made a lot of progress
 - 2D gesture recognition and messaging
 - 3D started, but not finished

2D gesture recognition

- Currently have 3 Gestures fully defined



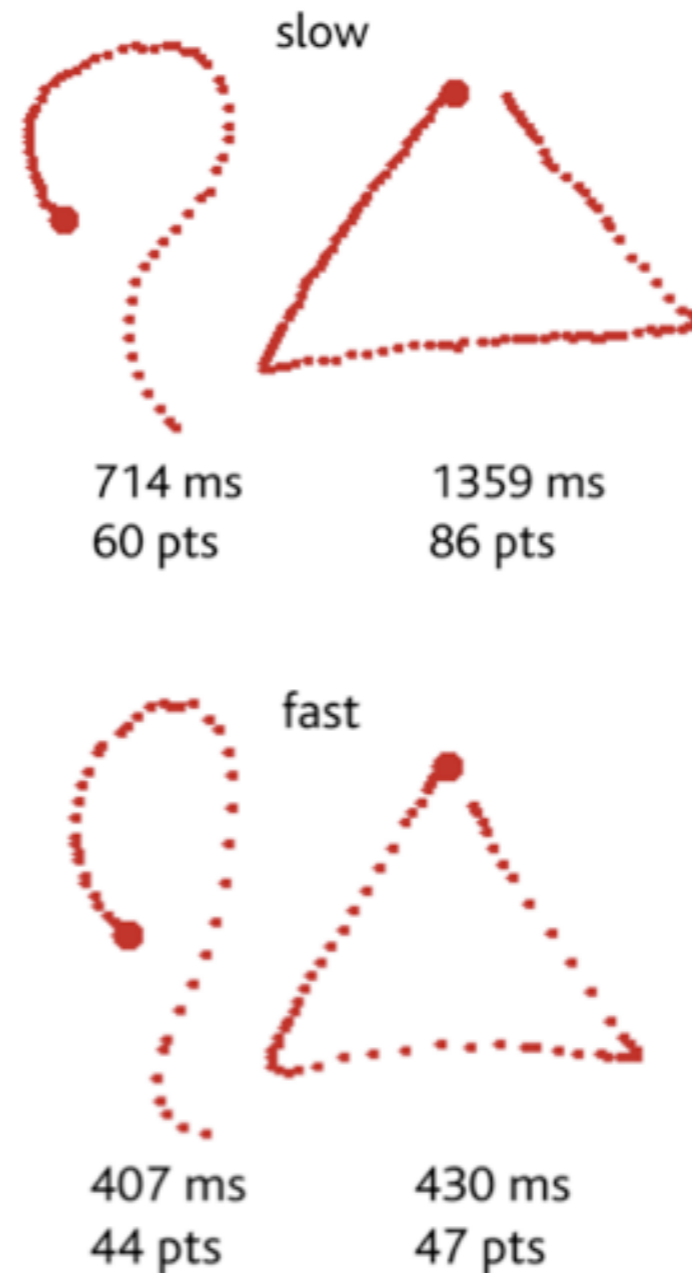
- Based on paper “\$1 gesture recognition
- Very simple to implement
- Good results, but small set and triangle only works one way!

\$ | Gesture recognition

1. Resample the points
2. Rotate based on the indicative angle
3. Scale and Translate
4. Find the optimal angle for the best score

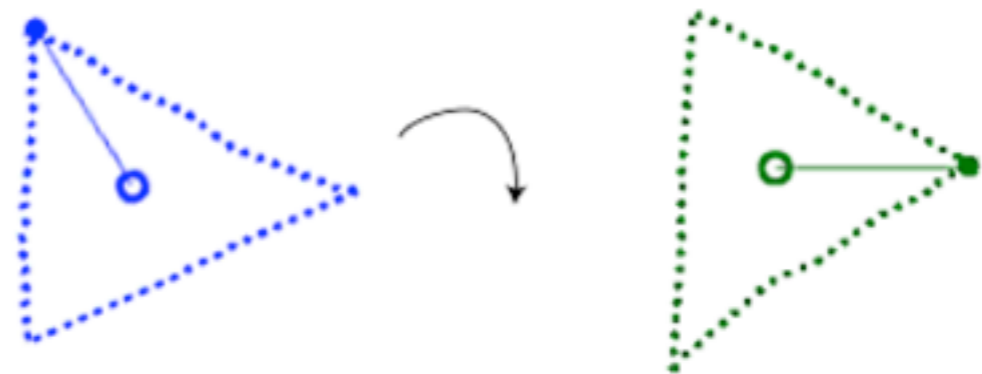
I. Resample the points

- Normalise the differences between gestures by resampling so both gestures have the same number of points.



2. Rotate based on indicative angle

- Take starting point and centroid of gesture.
- Defines 'angle' of gesture
- Normalise this so gestures are now rotationally invariant



3. Scale and Translate

- Centred and scaled to reference square of unit size.
- This is a non-uniform scaling...
- Puts gestures in the same context

4. Find the Optimal Angle

- Scoring stage
- Candidate gesture is compared with stored templates
- Minimise the 'path distance' between the gestures

\$3 recogniser

- Extend the 2D case to 3D
 - Accelerometer data much noisier than Touchscreen
- Exactly the same steps as before but now add in a heuristic (scoring scheme)

5. Scoring Heuristic

- Now produce a table of scores for each gesture template
 - ϵ is defined as the threshold score.
 - Iff the highest-scoring candidate in the score table has a score $> 1.1\epsilon$, return this candidate's gesture ID.
 - Iff, within the top three candidates in the score table, two candidates exist of the same gesture class and have a score $> 0.95\epsilon$, respectively, return the gesture ID of these two candidates.
 - Else, return "Gesture not recognized!".

Next steps..

- \$3 recogniser currently limited to highly defined gestures. i.e. 'press the button'
- Aim to find a way to segment the gestures so that the \$3 recogniser can be continuous
- Compare results with HMM implementation (Wii or Android based phone)