OctoPocus

Magdalena Keil and Jula McGibbon

Bauhaus Universität Weimar

2nd July 2016

Overview

Original Paper: Core Ideas

- User can execute gestures dynamically in different directions starting at the same point
- Learns to remember them after a few executions → "Learning by doing"
- Each gesture evokes a different command, e.g. Paste, Copy, etc.
- Combination of an on-touch feedforward and feedback system
- Less likely gesture paths become thinner or disappear entirely

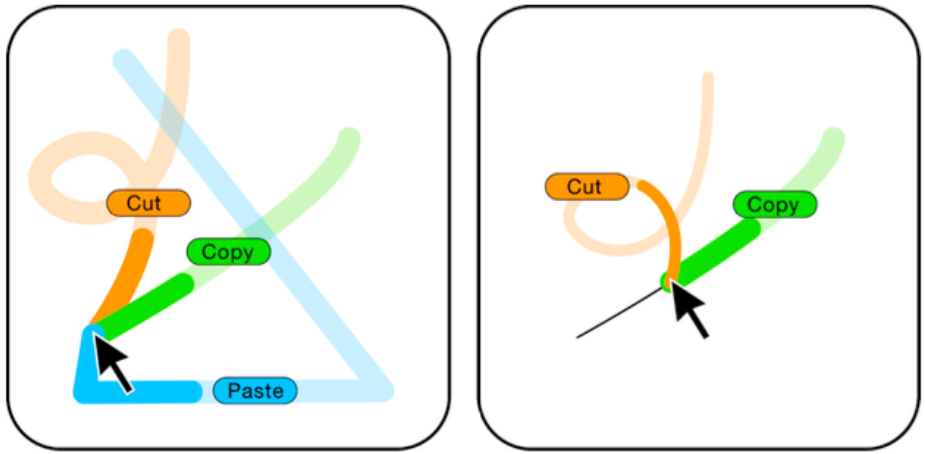
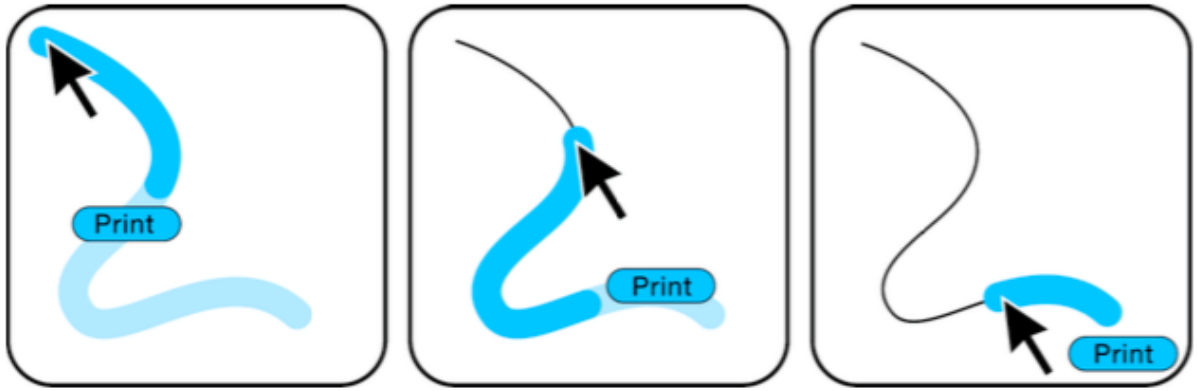


Figure: Left: Feedforward mechanism, Right: Feedforward and Feedback mechanism

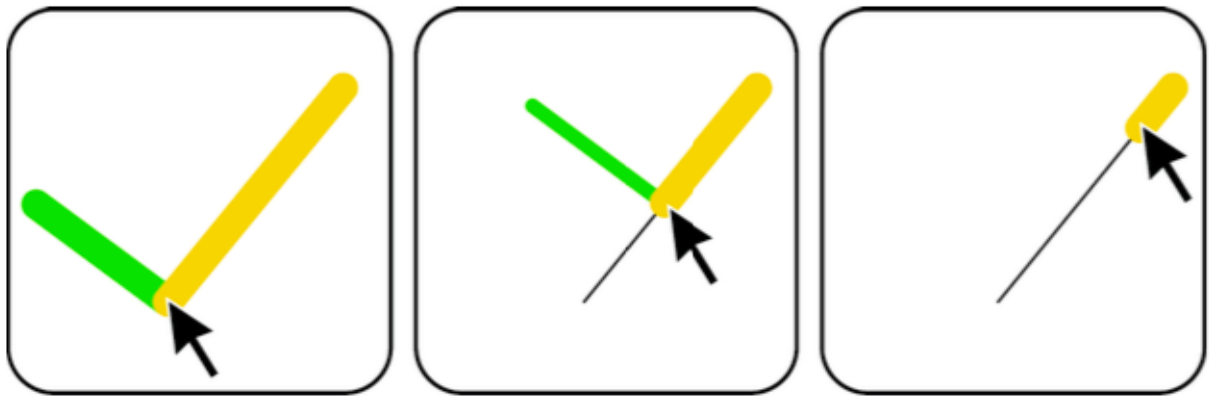
Feedforward mechanism

- Gives the user an impression of the gesture's shape and the related command
- Each template gesture has a prefix starting at the cursor and marking a small part of the whole gesture in a deep color, the rest has the same but translucent color
- The associated command is shown at the end of the prefix
- Disadvantage: Can take up a lot of screen space



Feedback mechanism

- Provides information about the recognition process after some action of the user
- Gesture recognition is accomplished by a classification algorithm using distance measures (Rubine's algorithm and Mahalanobis distance)
- The distances sum up to a consumable error rate until the input can no longer be recognized as a member of a given gesture class
- The consumable error rate is also mapped onto the thickness of the gesture



Novice or Expert?

Novice version:

- All paths are displayed by doing a long click

Expert version:

- No paths are displayed

Own Implementation: Similarities to the original implementation

- Gestures with different commands
- Feedforward: Prefix displays the next part to be drawn
- Feedback:
  - Already drawn path is shown in black
  - Distance measure (Euclidean distance) as recognition tool, also mapped onto the path's thickness
  - Likely paths have a fixed thickness
  - The less likely a path, the thinner it gets until it disappears

Own Implementation: Differences to the original implementation

- We are working with touch and not with a cursor
  $\Rightarrow$ Occlusion problems when paths are spread into all directions
- One path is provided for the user to create his own gesture paths, replacing it with the old ones
- No Novice or Expert mode yet

Own Implementation: Important Issues

- 1$ recognizer:
  - Was first considered for path recognition
  - Turned out to be complicated to work out with our purpose, since the figure could be drawn in every direction (even backwards) and was still recognized
- Finding the right thresholds:
  - Paths can be drawn quite close to each other
  - Solution: Doing error calculation only in the area close to the cursor
- Occlusion:
  - All paths are set by default into different directions
  - Solution: Default paths can be modified by the “New Path “function, e.g. if the user is right-handed, he can draw all paths from the lower right to the left, upperleft and top direction

The End