Pointing and Navigation

Michel Beaudouin-Lafon
Laboratoire de Recherche en Informatique
Université Paris-Sud / CNRS
mbl@lri.fr
http://insitu.lri.fr

Thanks to Yves Guiard for material on Fitts’ law

Outline

Pointing
Fitts’ law
Beating Fitts’ law
Multiscale pointing
More laws of movement

The importance of pointing

The most frequent action in Graphical User Interfaces
(together with entering text)

Many targets, some very small
e.g., pointing between the two ‘l’ in the word “small”
above

Screens are becoming larger

Pointing performance is limited by human capabilities,
not by the computer
If the computer knew where I want to point,
it could do it for me...

Fitts’ pointing paradigm

Seminal work by Paul Fitts in 1954
Speed-precision trade-off in directed movements

Initial hypothesis

\[ ID \text{ (bits)} = \log_2 \left( \frac{2D}{W} \right) \]

\[ MT = k \times ID \]

\[ ID = \text{Index of Difficulty} \]
\[ MT = \text{Movement Time} \]

If this proves true, \( \frac{ID}{MT} \text{ (bit/s)} = \text{constant} \)
This constant is the capacity of the human motor system
to transmit information (Shannon)
In practice... (Fitts’ original data)

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Task Conditions and Performance Data for a Biomechanical Tapping Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>A</td>
</tr>
<tr>
<td>3.00</td>
<td>5.50</td>
</tr>
<tr>
<td>1.00</td>
<td>3.00</td>
</tr>
<tr>
<td>2.00</td>
<td>5.50</td>
</tr>
<tr>
<td>3.00</td>
<td>5.50</td>
</tr>
</tbody>
</table>

Typical velocity profile

Several versions of Fitts’ law

Log version
Fitts (1954) \( MT = a + b \log_2(2D/W) \)
Mackenzie (1992) \( MT = a + b \log_2(D/W + 1) \)

Linear version
Schmidt et al. (1979) \( MT = a \cdot D/W \)

Power version
Meyer et al. (1988) \( MT = a \cdot (D/W)^{1/2} \)

In all cases, MT varies with the relative amplitude \( D/W \)
\( ID = f(D/W) \quad MT = a + b \cdot ID \)
Fitts’ law can be seen as a scale-invariance law
### Validity of Fitts’ law

- Fitts’ law is only valid within fairly small limits
- Absolute amplitude less than about one meter otherwise, there is a speed plateau
- Width larger than a fraction of a millimeter otherwise motor control is not precise enough
- Performance beyond those limits degrades quickly
- D/W is therefore bounded by about 2000, and so the ID (in the log formulation) is less than about 12

### Can we “beat” Fitts’ law?

- The index of performance IP = 1/b is about 10 bits/s in Fitts’ original experiment
- Pointing using a device (mouse, joystick, touchscreen…) has been shown to generally have a lower IP
- Research question: Can we use the computer to help us point faster?
- Other research question: Can we expand the limits of validity of Fitts’ law?

### Pointing in the wild

- Large collection of pointing data in the field
- 24 users, 2 million aimed movements, 1 billion pixels (352km)

### Improving pointing performance

- Idea 1: Reduce ID, i.e. decrease D and/or increase W
- Reducing distance: “drag’ n’ pop” (Baudisch)
- Reducing distance: “MAGIC pointing” (Zhai)
  - Track eye-gaze to teleport cursor close to the target
Improving pointing performance

Increasing target size: auto-expansion (McGuffin)
Expand potential targets when the cursor approaches them

Performance predicted by expanded target size (not original size)
BUT: does not work in the Mac OS X dock because adjacent targets move -> expansion cannot work with dense targets

Increasing size: semi-infinite targets
Pointing on the side of the screen

Edging is closest to semi-infinite pointing (Appert)

Improving pointing performance

Idea 2: Increase maximal speed
Manipulate the "control-display gain", i.e. the ratio between the motion of the device and the corresponding motion of the cursor "Mouse acceleration"

Effect of dynamic gain on pointing performance (Casiez)

Semantic pointing (Blanch)
Each target has a visual size and a motor size
Cursor moves faster between targets, and slows down when approaching a target

Sample applications:
Object pointing (Guiard)
Skip empty space: pointing in constant time! (in theory...)

(c) 2011, Michel Beaudouin-Lafon, mbl@lri.fr
Improving performance

Bubble cursor (Grossman): best technique known today
Combines area cursors, object pointing and target expansion
The cursor always designates the closest target

Dynasplot (Chapuis): combines bubble cursor with regular cursor to point in empty space

Improving pointing performance

Are we done yet?
NO!

Two categories of approaches:
target-agnostic: do not need to know where targets are
target-aware: needs to know potential targets

Target-aware techniques are more efficient,
but it is often difficult to know what the targets are

Probabilistic approaches: learn targets and user’s habits

Breaking the limits of Fitts’ law

Fitts’ law is valid only for ID < 12 bits, D < 1m, W > 0.5mm

These physiological limits can be overcome in an information world that supports zooming
Zooming in: small targets become bigger
Zooming out: large amplitudes become smaller

What is the performance of pointing in a zoomable world?

Zoomable User Interfaces

Pad (Perlin & Fox)
Space-scale diagrams (Furnas & Bederson)

Represent scale as a vertical dimension

Zooming = moving the viewing window up and down
The size of the viewing window is fixed

Multiscale pointing (Guiard & Beaudouin-Lafon)

Pointing in a zoomable world requires navigation:
- Zoom out to get the target in view
- Pan to put the target in the center
- Zoom in to enlarge the target (pan to adjust)

Effect of view size on pointing performance

Effect of view size: $MT = k \frac{ID}{V}$
But only up to a certain threshold for $V$

Orthozoom (Appert & Fekete)

Extend scrollbar to pan and zoom 1D documents
Use orthogonal dimension to zoom

Orthozoom is twice as fast as the best known technique: Speed-Dependent Automatic Zooming (SDAZ)
## Other laws of movement

Generalizing Fitts’ law to 2D pointing

\[
\begin{align*}
\log_{10}(D) &= \log_{2}\left(\frac{D}{W} + 1\right) \\
\log_{10}(D_{\text{eff}}) &= \log_{2}\left(\frac{D}{W_{\text{eff}}} + 1\right)
\end{align*}
\]

Goal-passing / crossing (Accot & Zhai)

\[
\log_{10}(D) = \log_{2}\left(\frac{D}{W} + 1\right)
\]

Steering law (Accot)

Tunnel: \[ MT = a + \frac{b}{W} \]

General case: \[ T_C = a + b \int \frac{d\rho}{W(\rho)} \]

## Crossy – a crossing-based interface (Apitz)

### Conclusion

Basic interactions such as pointing are still far from optimal

Fitts’ law is a surprisingly robust law

Information is key:
- Information available in the display
- Information perceived by the user
- Information produced by the motor system
- Information captured by the system