

Pointing and Navigation

Michel Beaudouin-Lafon

Laboratoire de Recherche en Informatique

Université Paris-Sud / CNRS

mbl@lisn.fr

<http://ex-situ.lri.fr>

Thanks to Yves Guiard for material on Fitts' law

Outline

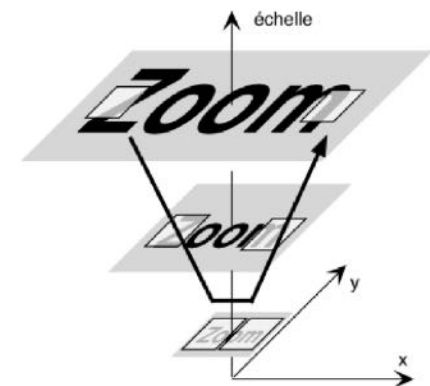
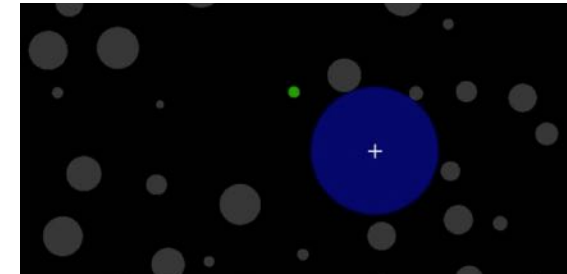
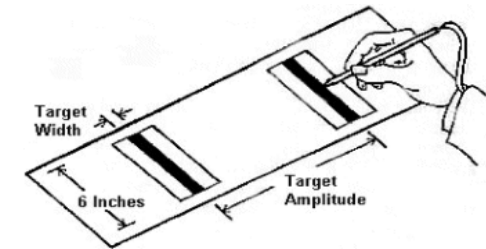
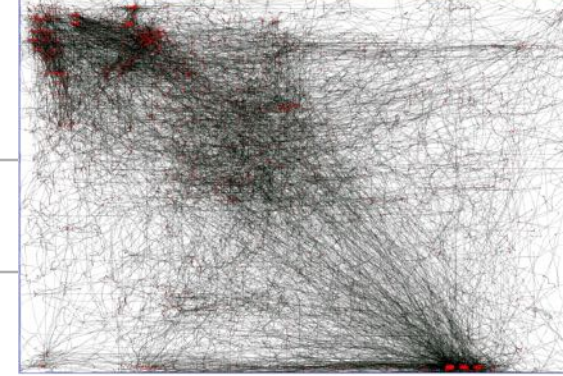
Pointing

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 ... and smaller

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

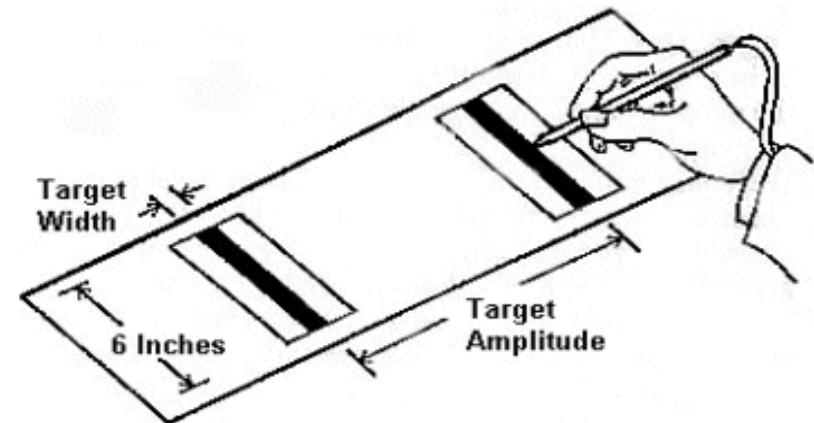
Initial hypothesis

$$ID \text{ (bits)} = \log_2 (2D/W)$$

$$MT = k * ID$$

ID = Index of Difficulty

MT = Movement Time



If this proves true, ID/MT (bit/s) = constant

This constant is the capacity of the human motor system to transmit information (Shannon)

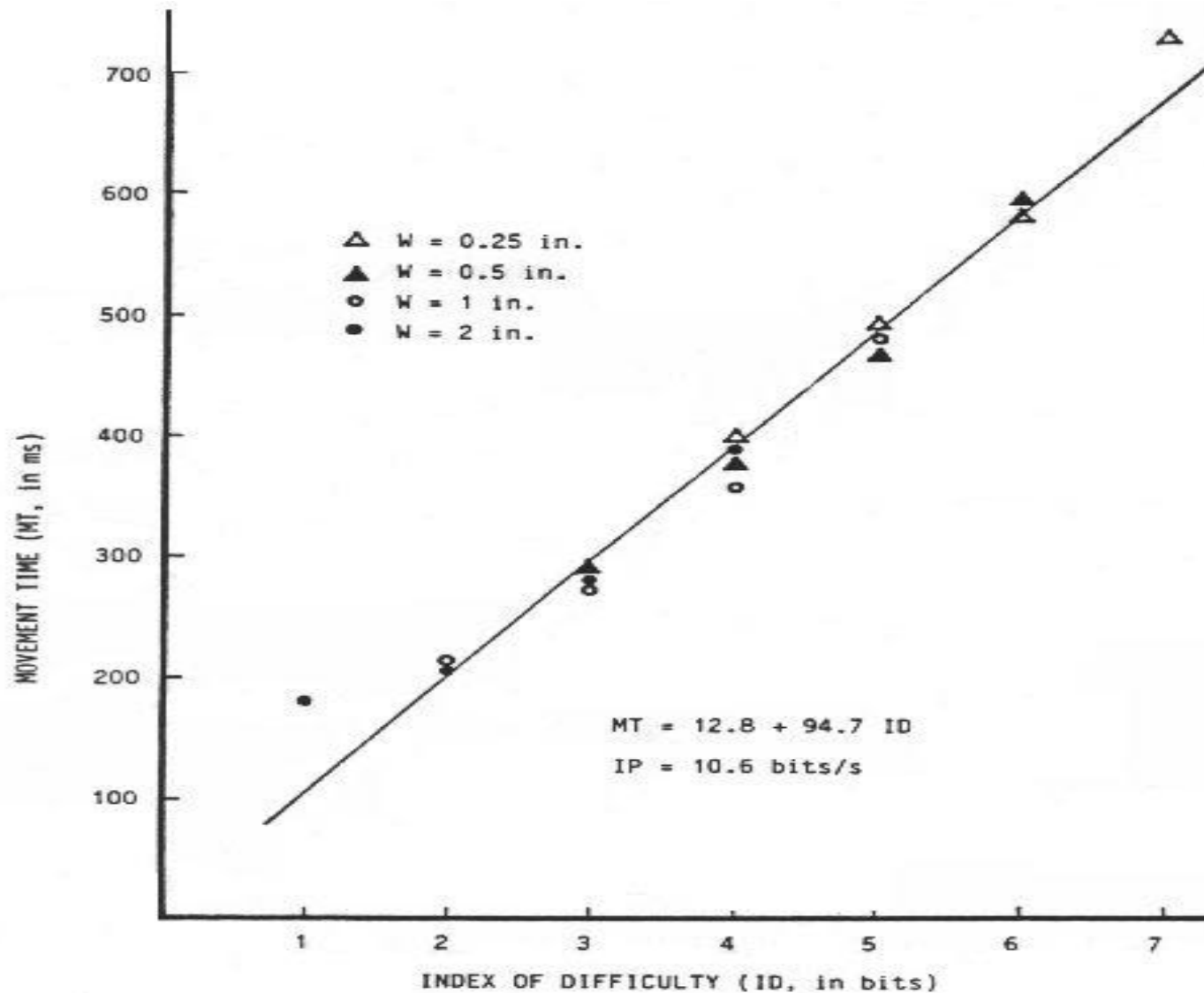
In practice... (Fitts' original data)

TABLE 1
TASK CONDITIONS AND PERFORMANCE DATA FOR 16 VARIATIONS OF A
RECIPROCAL TAPPING TASK
($N =$ the same 16 Ss at each condition)

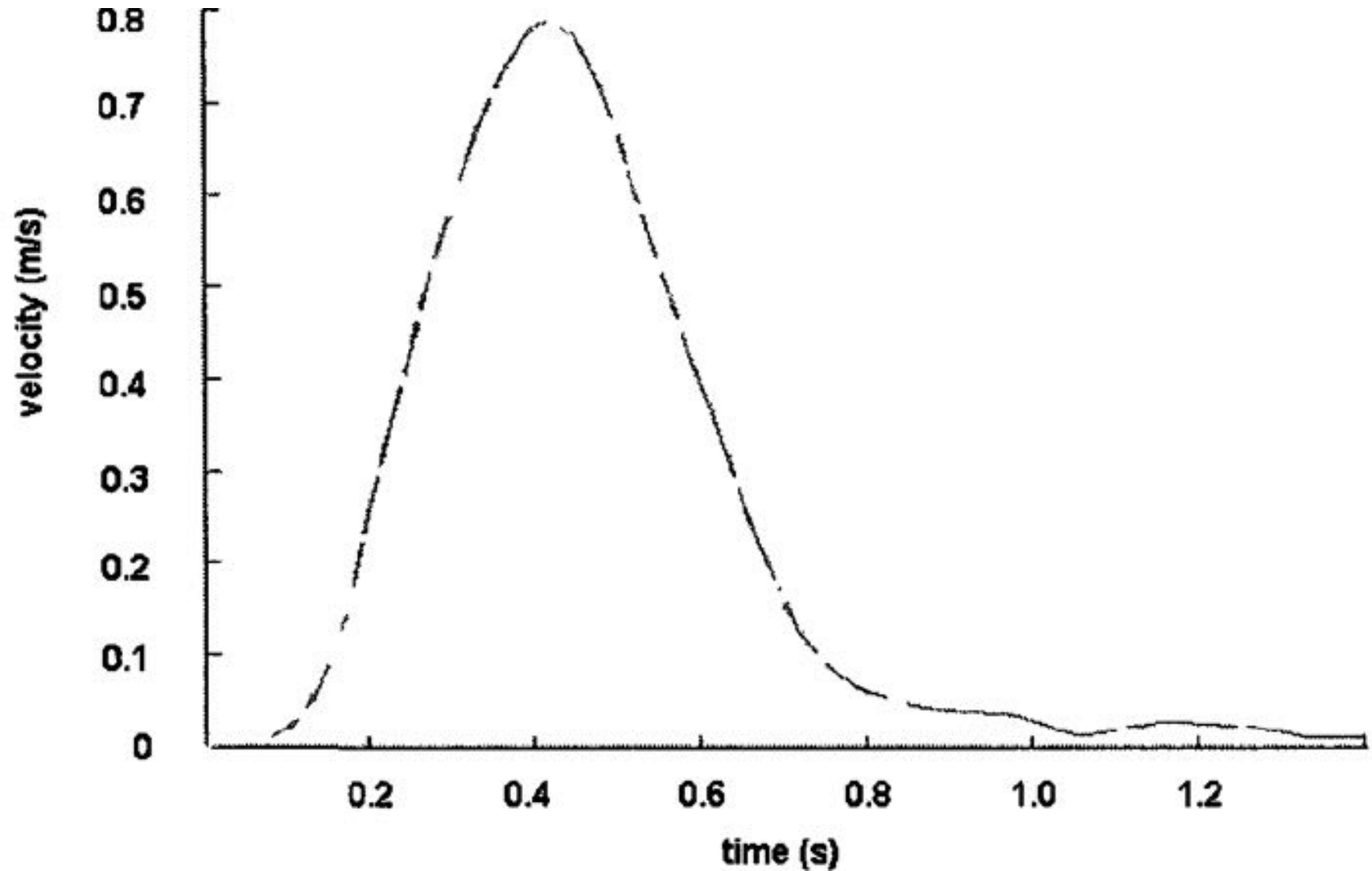
Tolerance and Amplitude Conditions			1-oz. Stylus				1-lb. Stylus			
W_s	A	I_d	t	Errors (%)	I_p	Rank	t	Errors (%)	I_p	Rank
.25	2	4	.392	3.35	10.20	11	.406	3.80	9.85	7
.25	4	5	.484	3.41	10.33	9	.510	3.83	9.80	8
.25	8	6	.580	2.78	10.34	8	.649	4.04	9.24	13
.25	16	7	.731	3.65	9.58	14	.781	4.08	8.96	15
.50	2	3	.281	1.99	10.68	5	.281	0.88	10.68	4
.50	4	4	.372	2.72	10.75	3.5	.370	2.16	10.81	2
.50	8	5	.469	2.05	10.66	6	.485	2.32	10.31	6
.50	16	6	.595	2.73	10.08	12	.641	2.27	9.36	11
1.00	2	2	.212	0.44	9.43	15	.215	0.13	9.30	12
1.00	4	3	.260	1.09	11.54	1	.273	0.85	10.99	1
1.00	8	4	.357	2.38	11.20	2	.373	1.17	10.72	3
1.00	16	5	.481	1.30	10.40	7	.526	1.32	9.50	10
2.00	2	1	.180	0.00	5.56	16	.182	0.00	5.49	16
2.00	4	2	.203	0.08	9.85	13	.219	0.09	9.13	14
2.00	8	3	.279	0.87	10.75	3.5	.284	0.65	10.56	5
2.00	16	4	.388	0.65	10.31	10	.413	1.72	9.68	9

Note.— W_s is the width in inches of the target plate. A is the distance in inches between the centers of the two plates. t is the average time in seconds for a movement from one plate to the other. The performance index, I_p , is discussed in the text.

In practice... (plot of Fitts' original data by Mackenzie)



Typical velocity profile



Several versions of Fitts' law

Log version

Fitts (1954) $MT = a + b \log_2(2 \mathbf{D/W})$

Mackenzie (1992) $MT = a + b \log_2(\mathbf{D/W} + 1)$

Linear version

Schmidt et al. (1979) $MT = a * \mathbf{D/W}$

Power version

Meyer et al. (1988) $MT = a (\mathbf{D/W})^{1/2}$

In all cases, MT varies with the relative amplitude D/W

$$ID = f(\mathbf{D/W}) \quad MT = a + b * ID$$

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

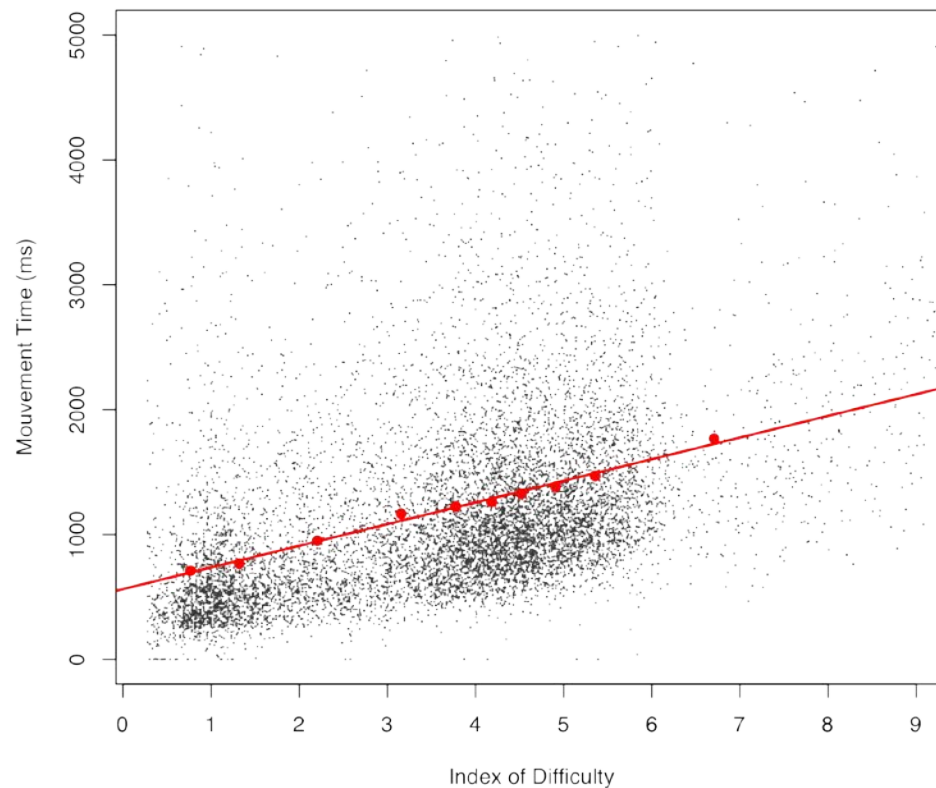
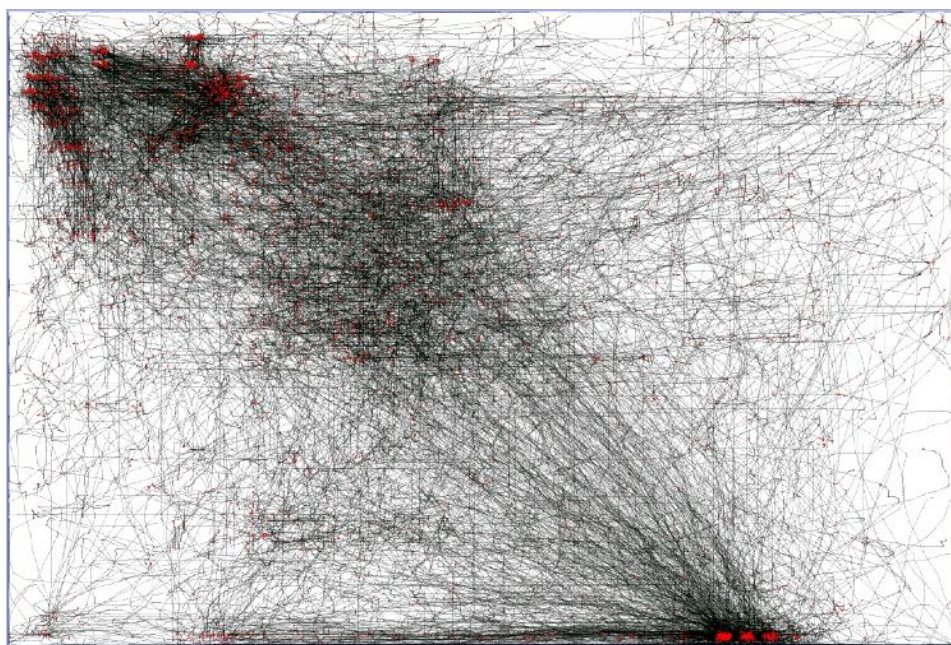
Performance beyond those limits degrades quickly

D/W is therefore bounded by about 2000,
and so the ID (in the log formulation) is less than about 12

Pointing in the wild

Large collection of pointing data in the field

24 users, 2 million aimed movements, 1 billion pixels (352km)



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?

Improving pointing performance

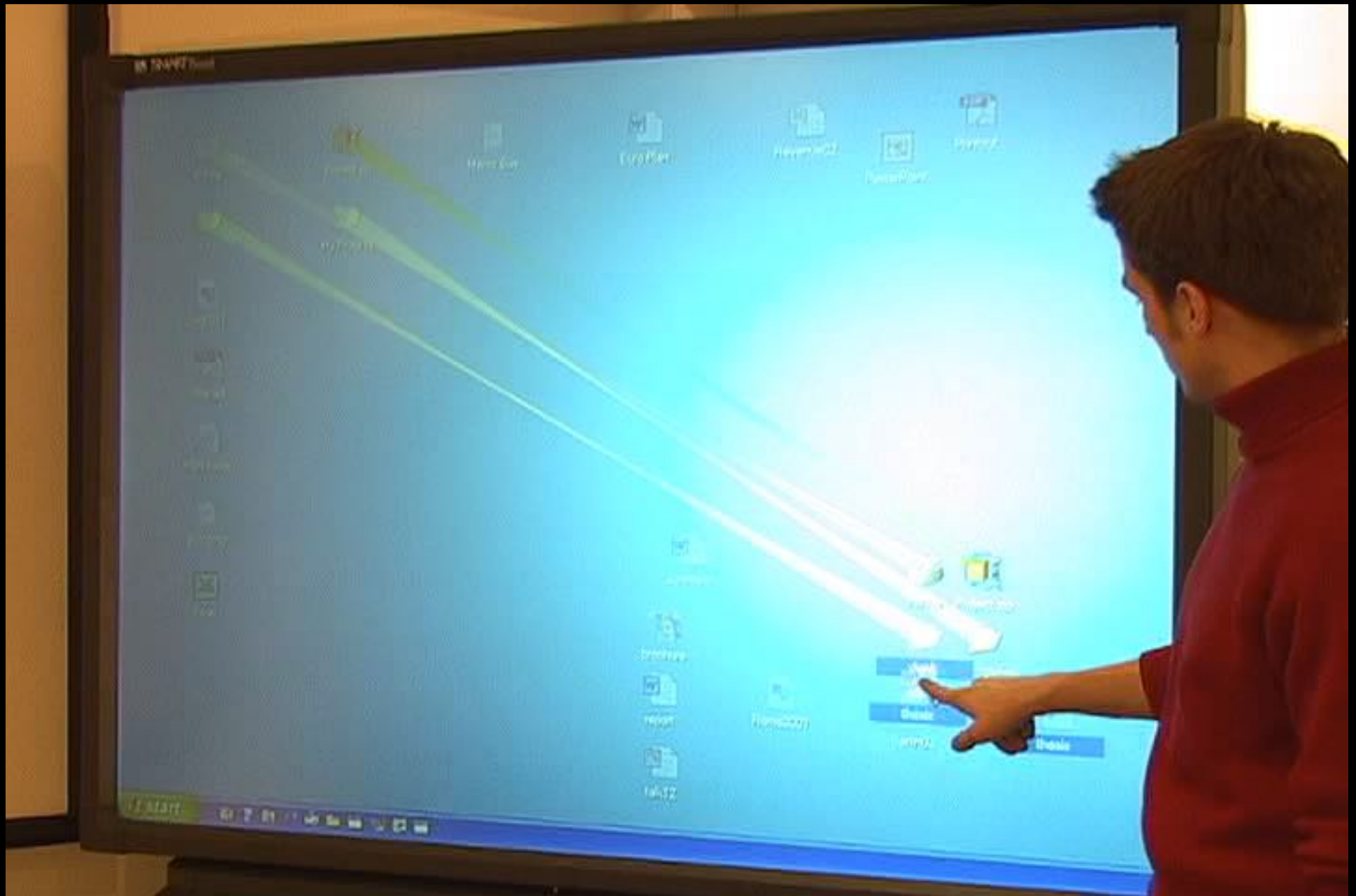
Idea 1: Reduce ID, i.e. decrease D and/or increase W

Reducing distance: “drag’n’pop” (Baudisch et al.)



Reducing distance: “MAGIC pointing” (Zhai et al.)

Track eye-gaze to teleport cursor close to the target



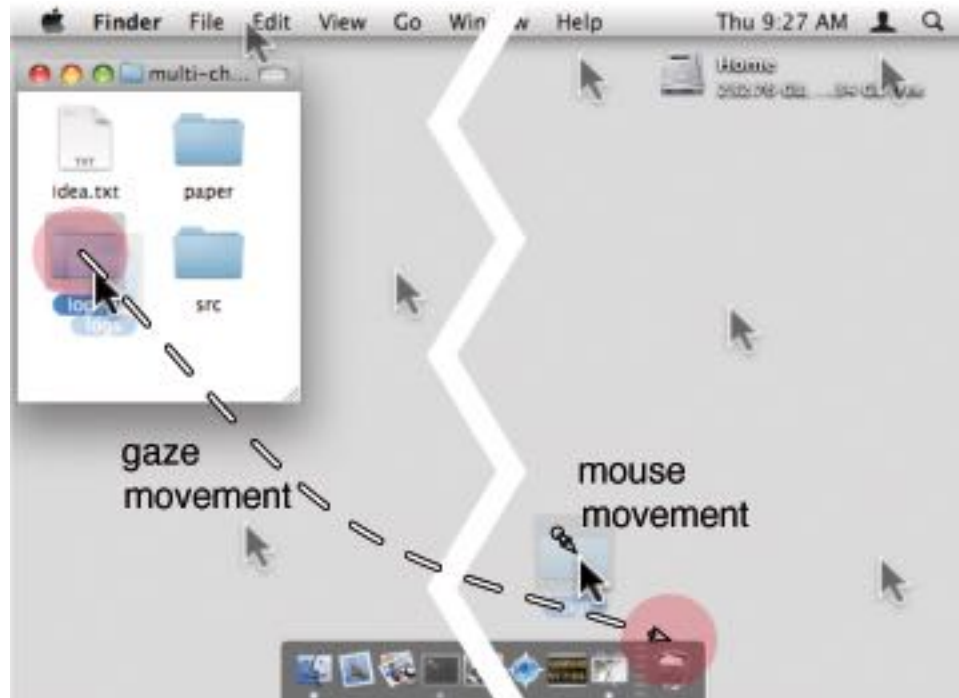
Drag-and-pop - Baudisch et al., 2003

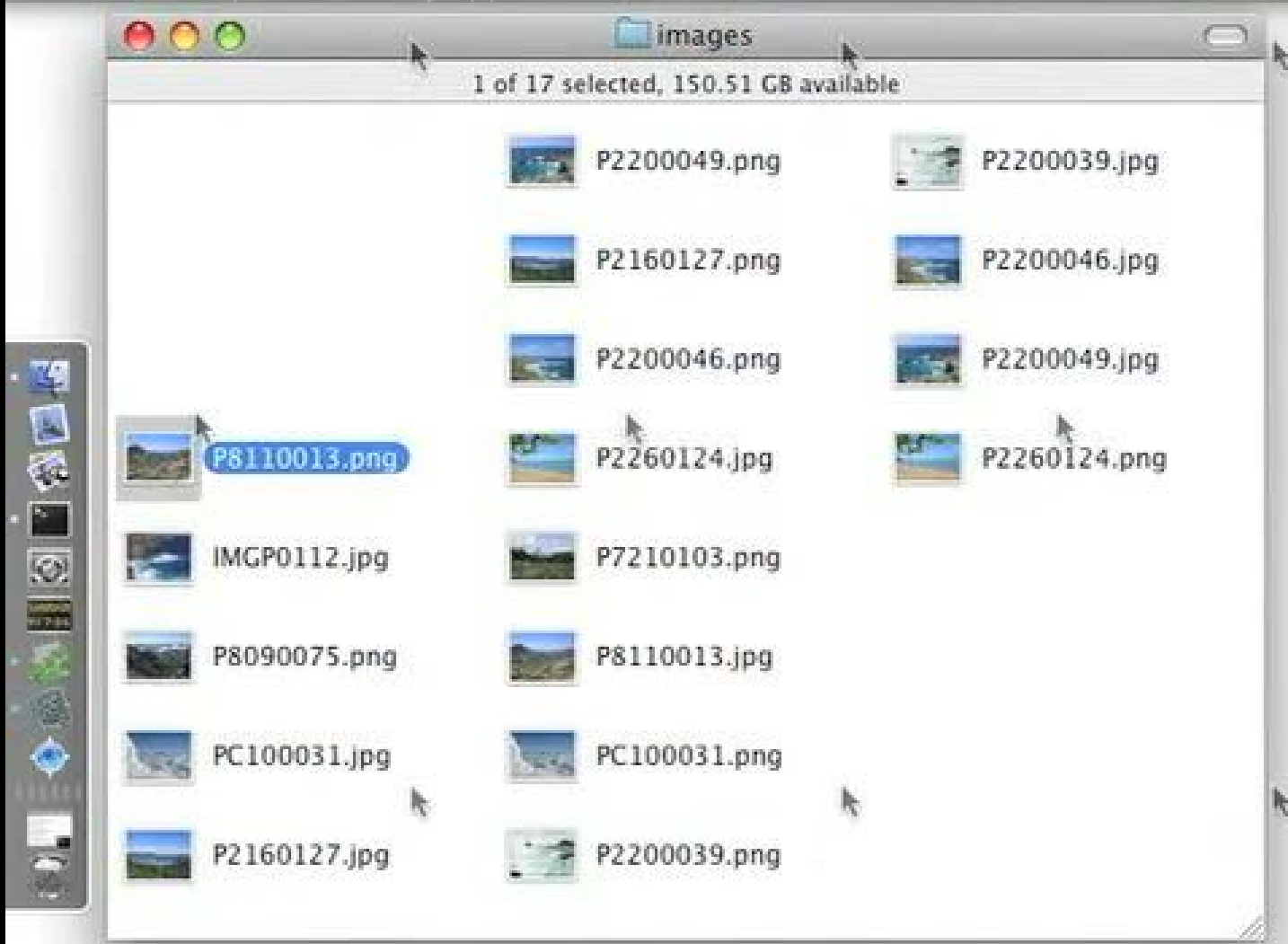
Improving pointing performance

Idea 1: Reduce ID, i.e. decrease D and/or increase W

Reducing distance: “rake cursor” (Blanch & Ortega)

Multiple parallel cursors + eye tracking





Improving pointing performance

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



Performance predicted by expanded target size

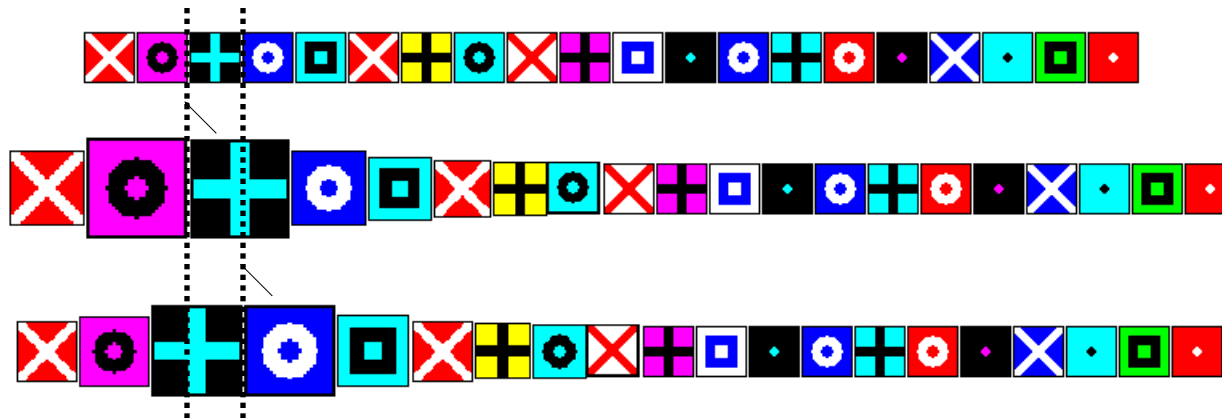
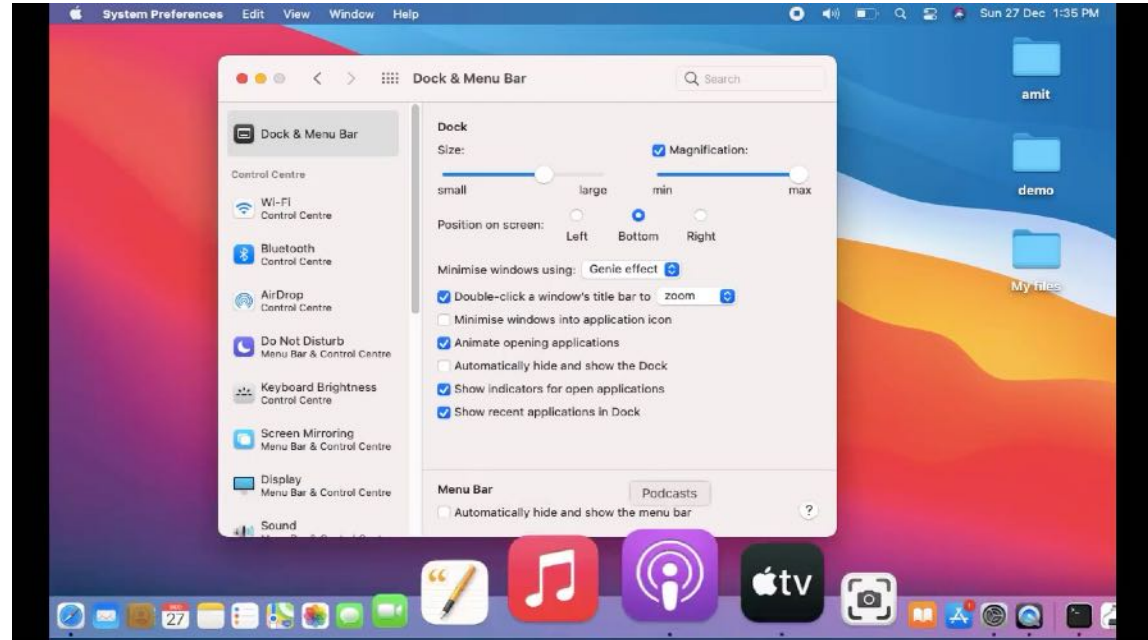


Even if the user does not expect the expansion (Zhai et al.)

Improving pointing performance

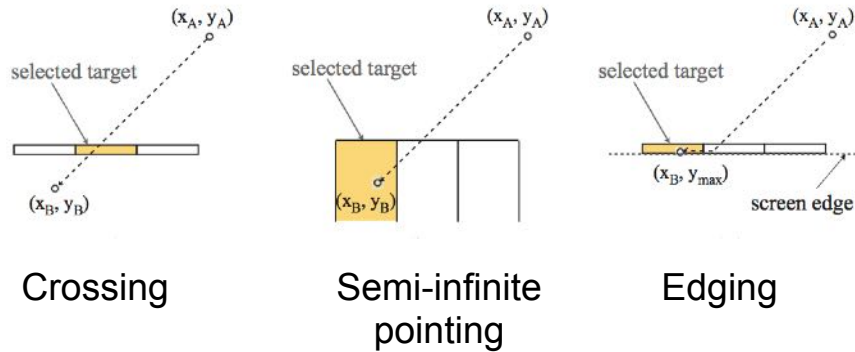
Application to the
MacOS Dock

BUT
it does not work!

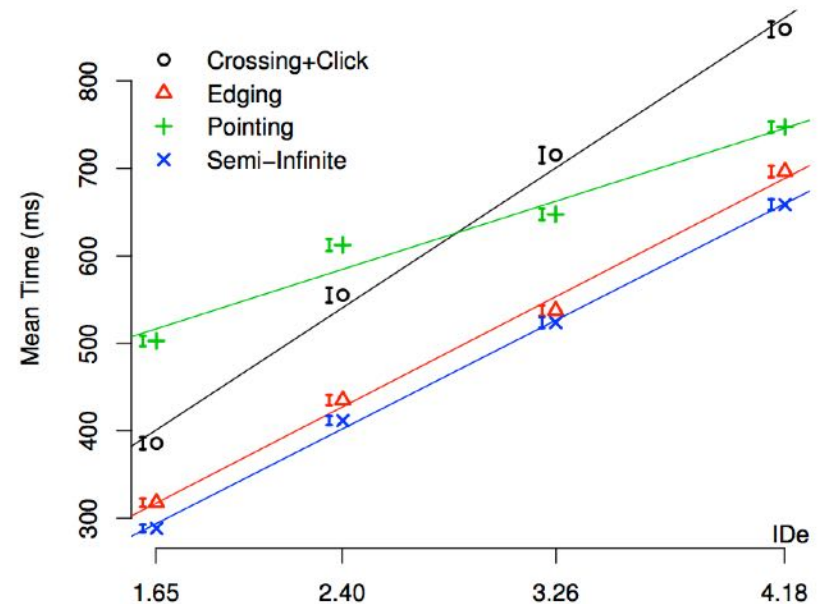
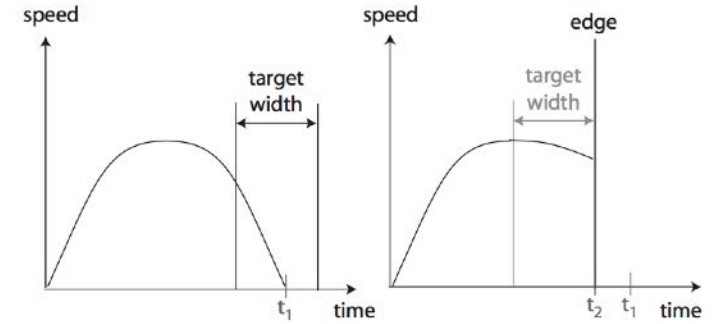


Improving pointing performance

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



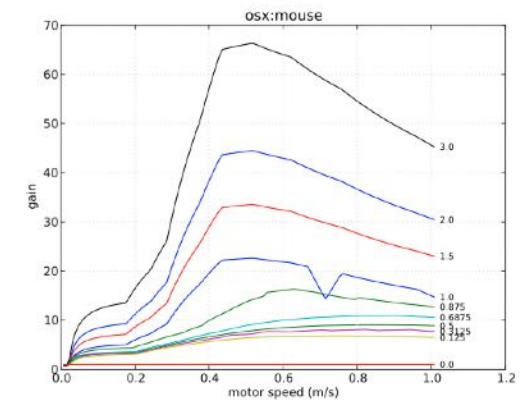
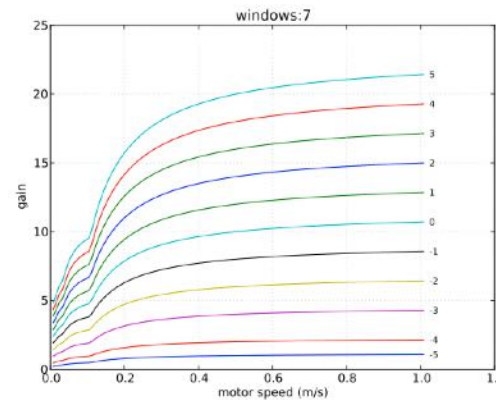
Edging is closest to semi-infinite pointing (Appert et al.)



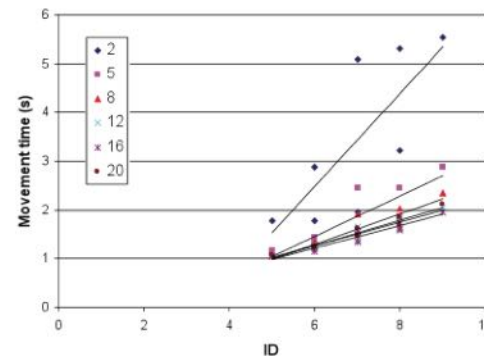
Improving pointing performance

Idea 2: Increase maximal speed

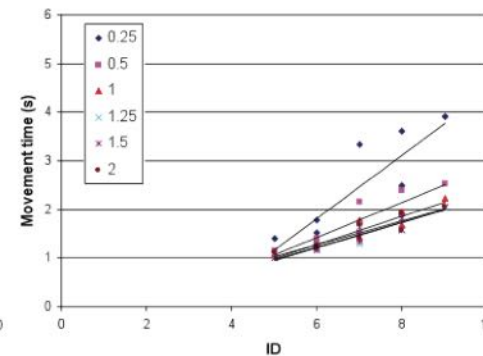
Manipulate the “control-display gain”, i.e. the ratio between the motion of the device and the corresponding motion of the cursor
 “Mouse acceleration”



Effect of dynamic gain on pointing performance (Casiez)



(a) Constant Gain



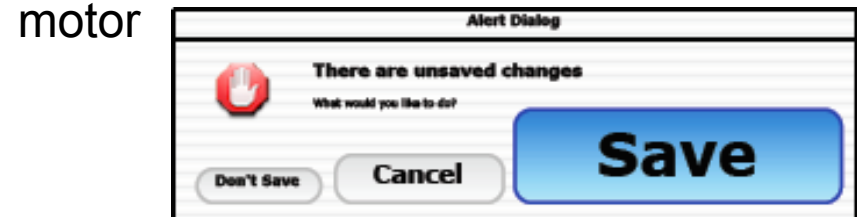
(b) Pointer Acceleration

Improving pointing performance

Semantic pointing (Blanch et al.)

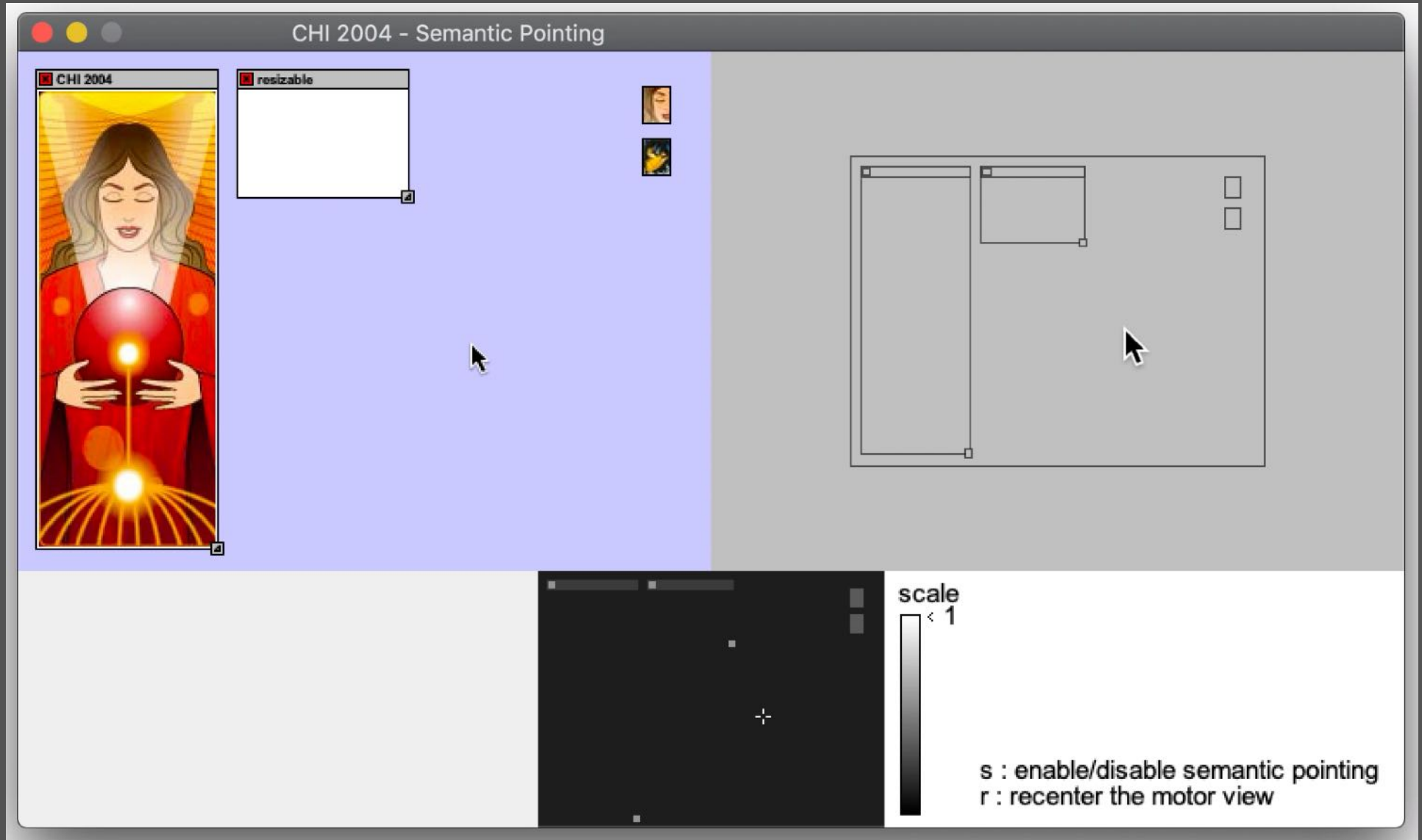
Each target has a visual size and a motor size
 Cursor moves faster between targets,
 and slows down when approaching a target

Sample applications:



Object pointing (Guiard et al.)

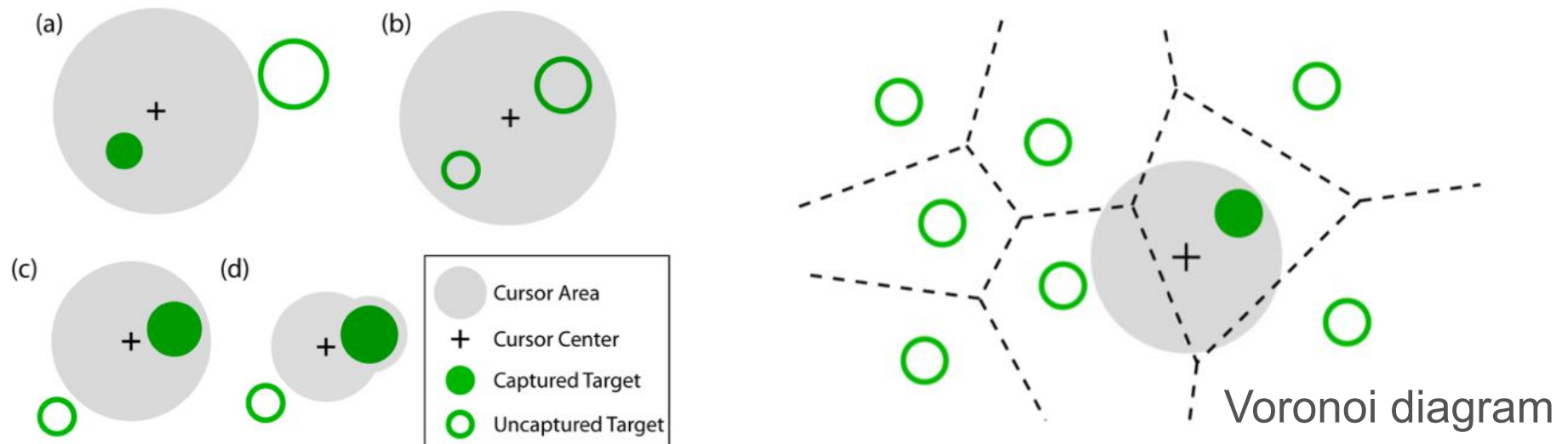
Skip empty space: pointing in constant time! (in theory...)



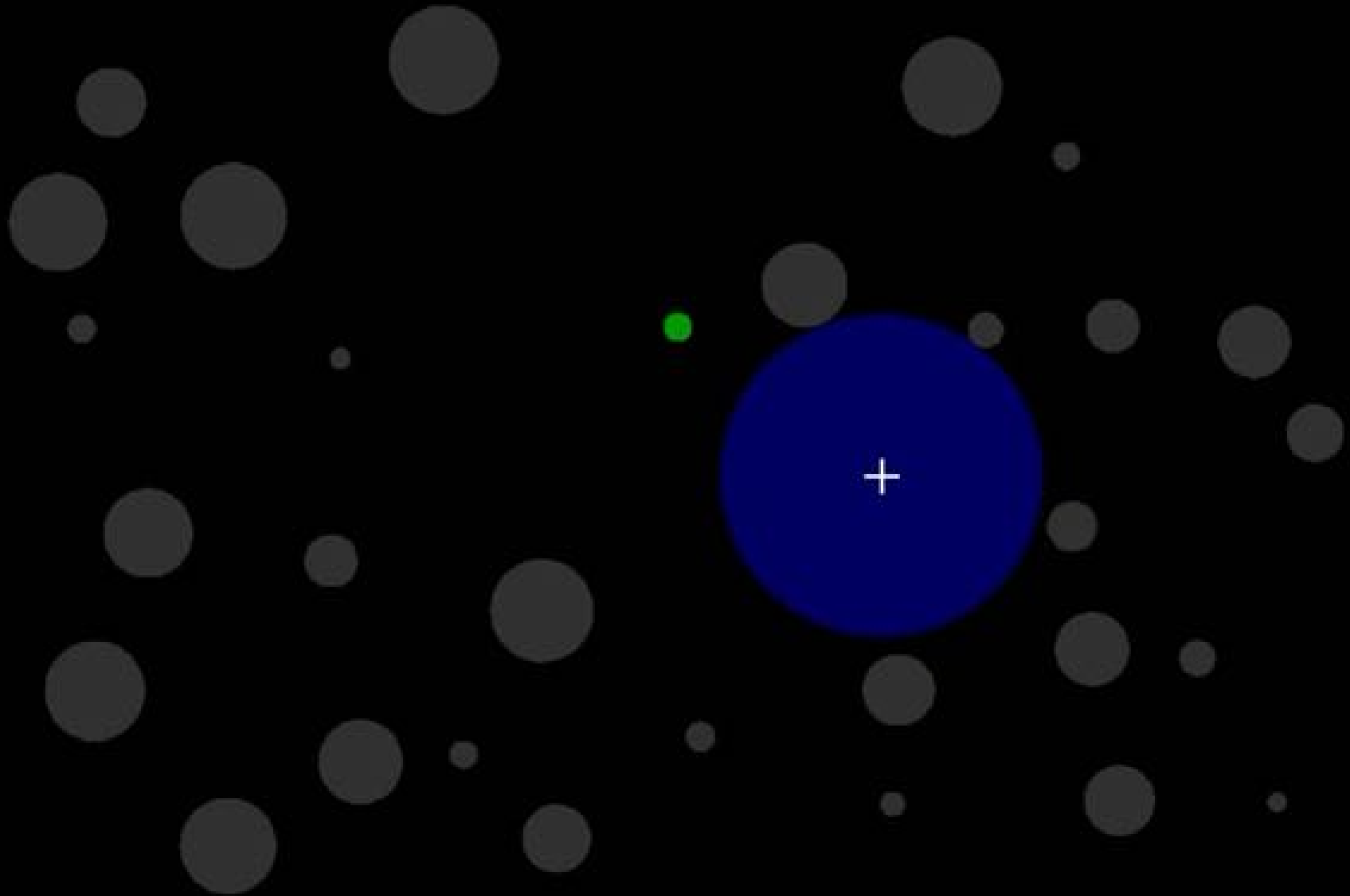
Semantic Pointing - Guiard, Blanch & Beaudouin-Lafon, 2004

Improving performance

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



Dynaspot (Chapuis et al.): combines bubble cursor with regular cursor to point in empty space



Bubble Cursor - Grossman & Balakrishnan, 2005



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, $D < 1m$, $W > 0.5mm$

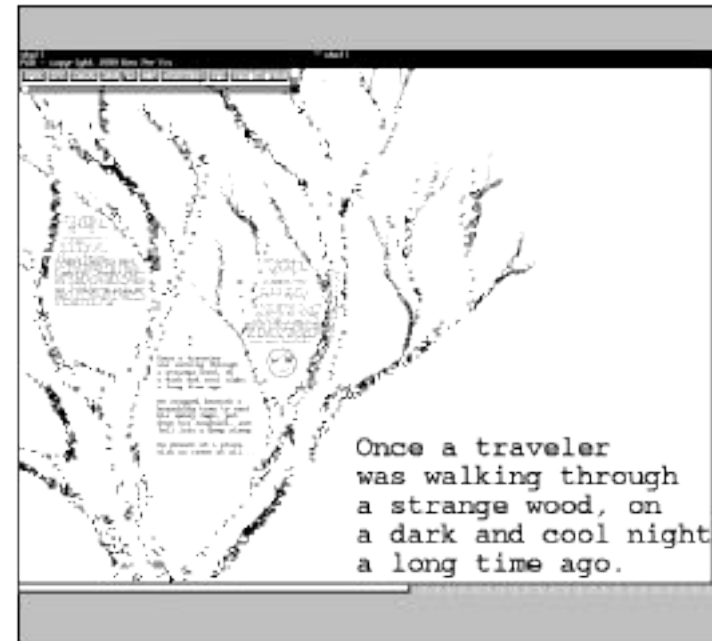
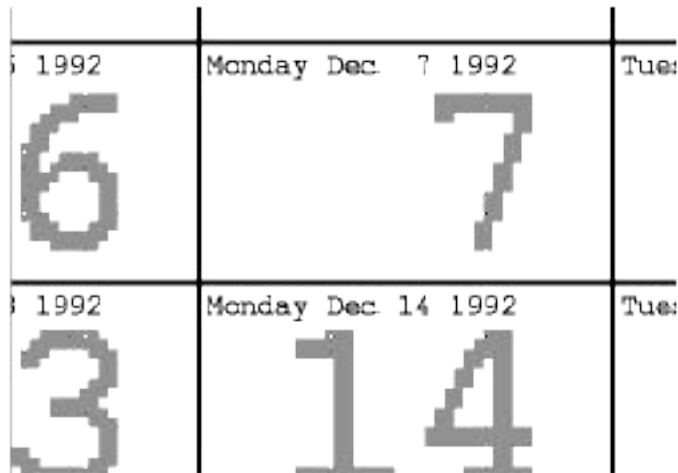
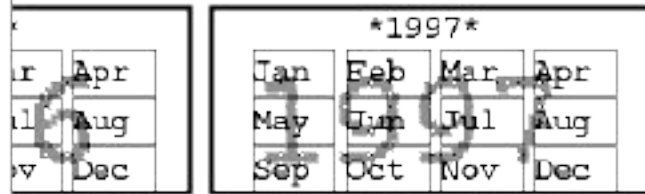
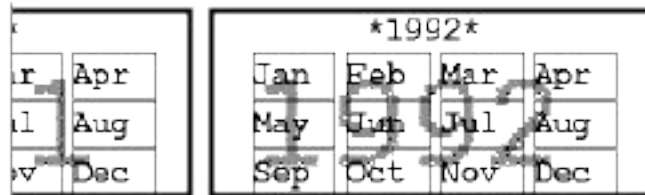
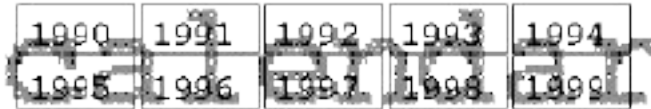
These physiological limits can be overcome in an information world that supports zooming

Zooming in: small targets become bigger

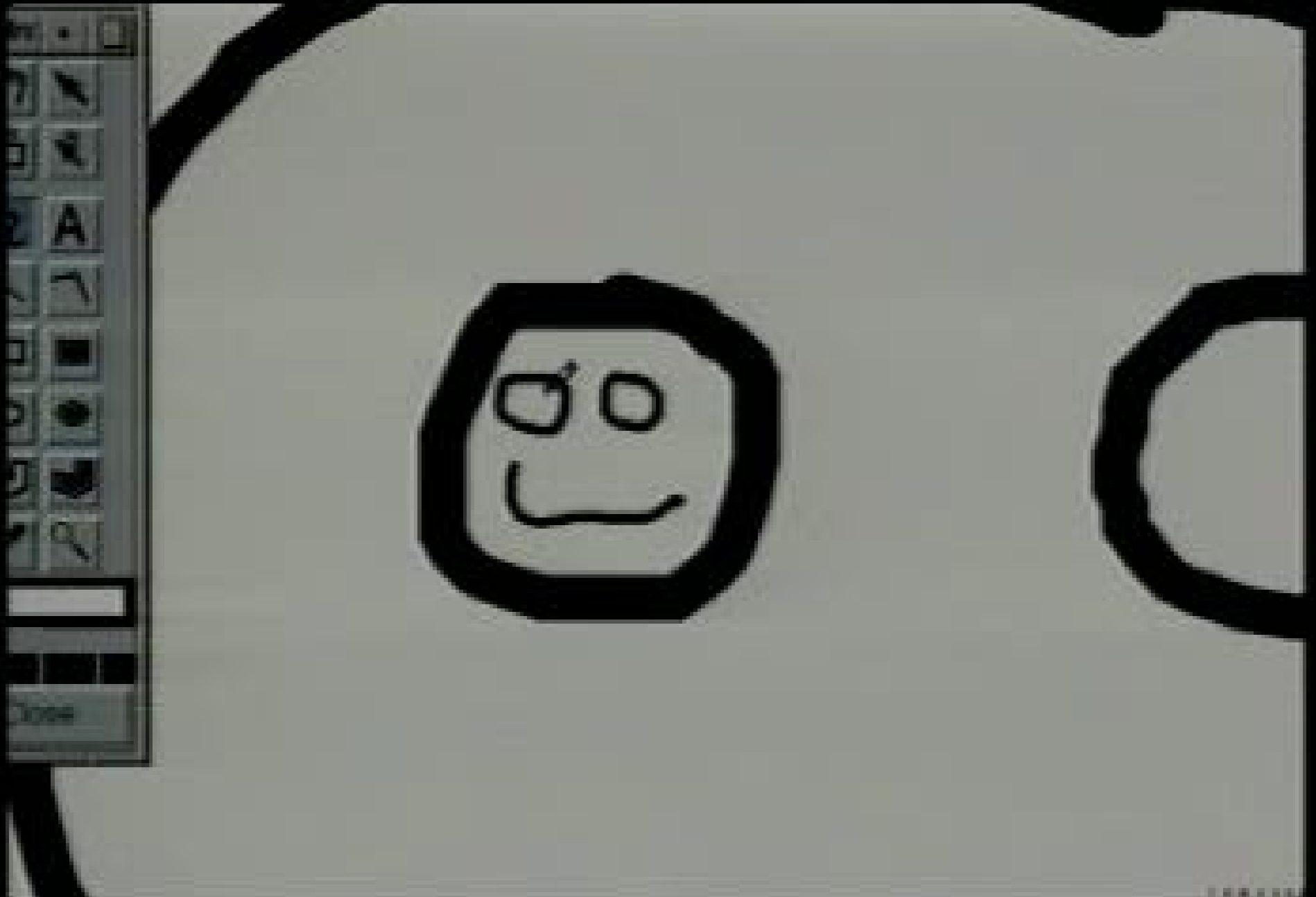
Zooming out: large amplitudes become smaller

What is the performance of pointing in a zoomable world?

Zoomable User Interfaces



Pad (Perlin & Fox)

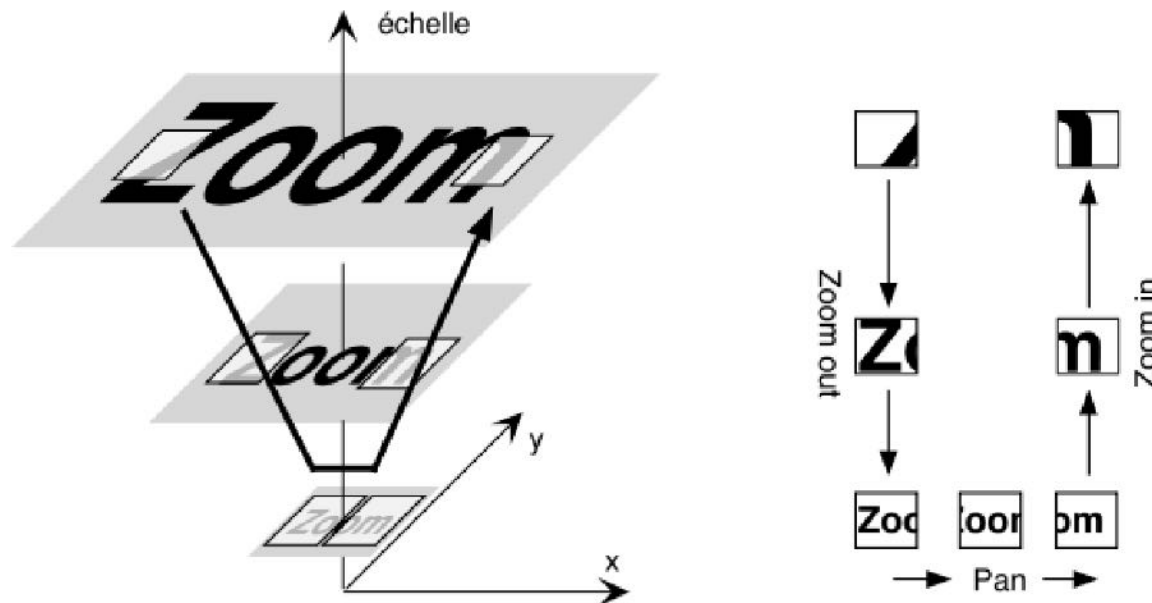


Space-scale diagrams (Furnas & Bederson)

Represent scale as a vertical dimension

Zooming = moving the viewing window up and down

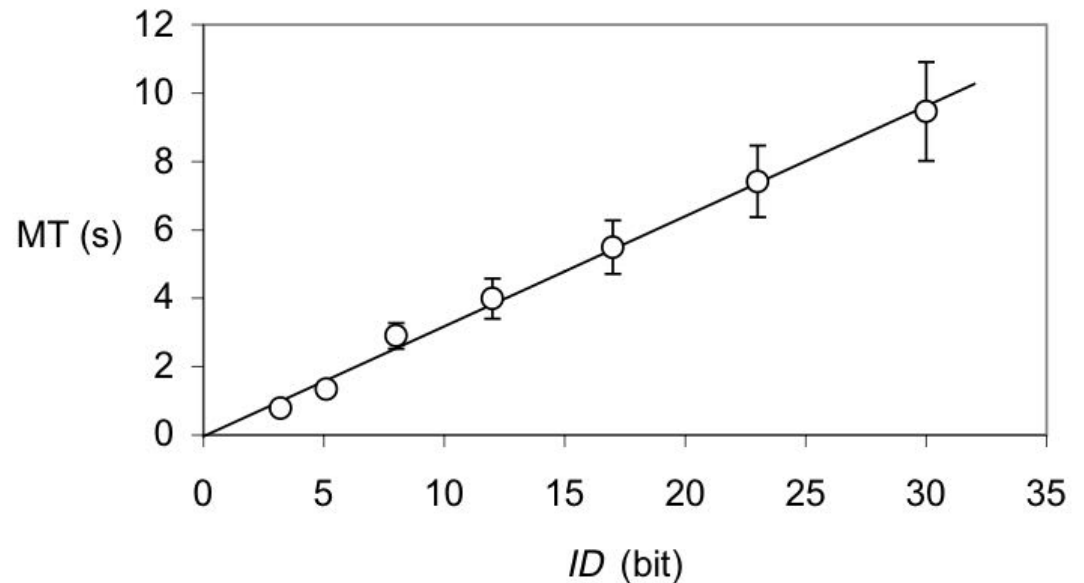
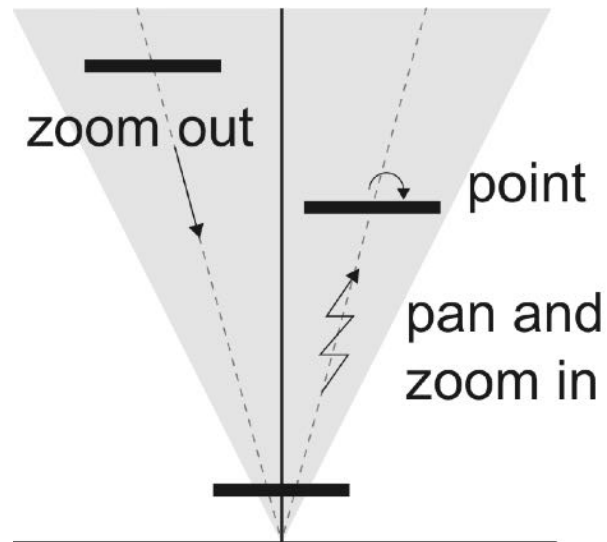
The size of the viewing window is fixed



Multiscale pointing (Guiard & Beaudouin-Lafon)

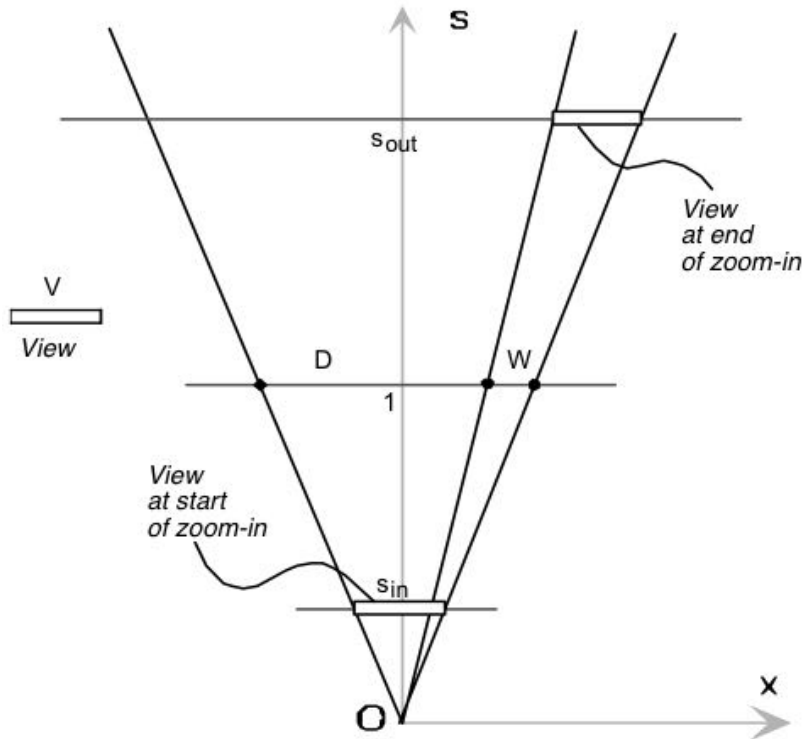
Pointing in a zoomable world requires navigation:

- 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)

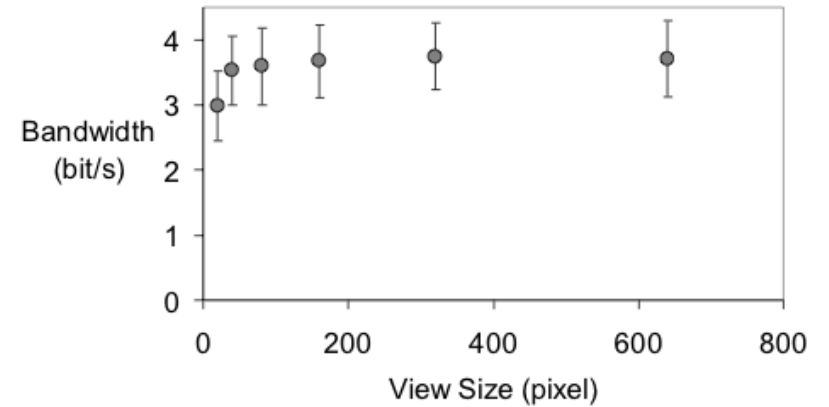


Effect of view size on pointing performance

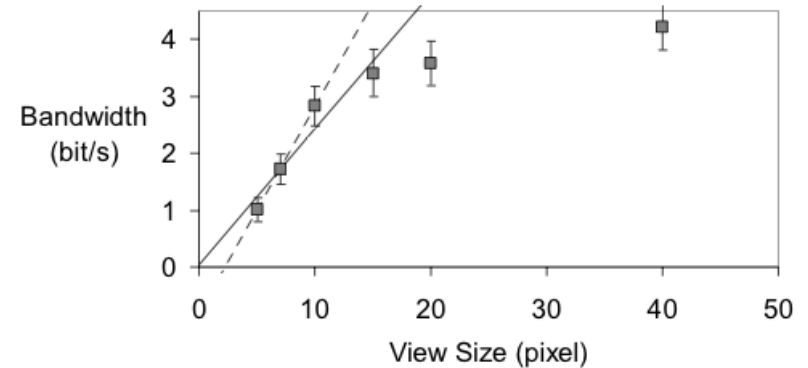
Effect of view size: $MT = k ID / V$
 But only up to a certain threshold for V



Exp. 1

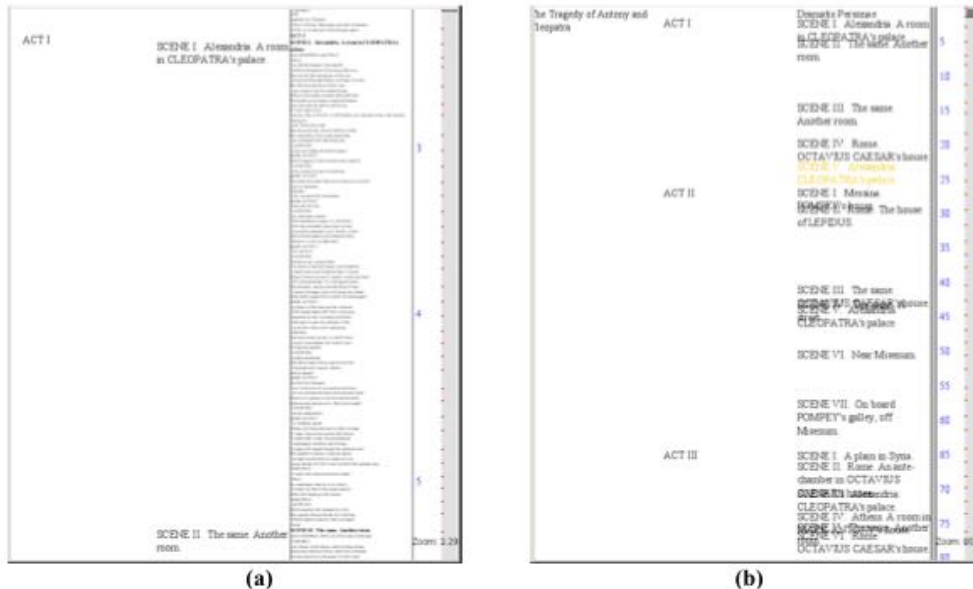


Exp. 2



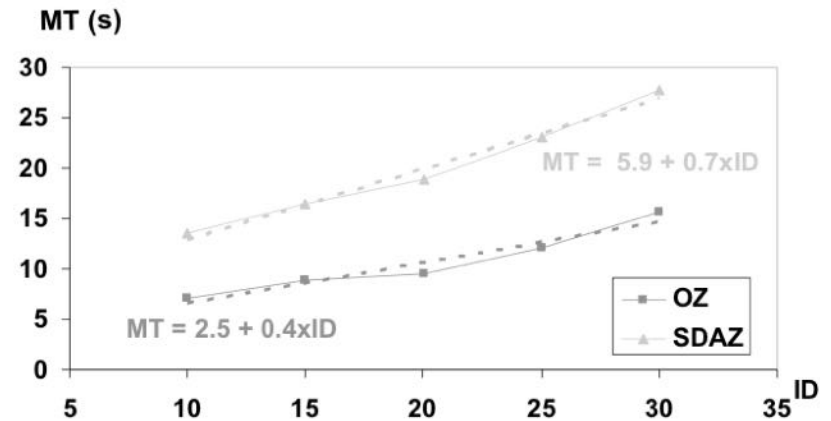
Orthozoom (Appert & Fekete)

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



Navigating the plays of Shakespeare

Orthozoom is twice as fast as the best know technique: Speed-Dependent Automatic Zooming (SDAZ)

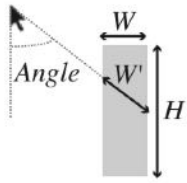


The Tragedy of Antony and Cleopatra	ACT I	
	ACT II	20
	ACT III	40
	ACT IV	80
	ACT V	100
All's Well That Ends Well	ACT I	120
	ACT II	140
	ACT III	160
	ACT IV	180
	ACT V	200
As You Like It	PROLOGUE	220
	ACT II	240
	ACT III	260
	ACT IV	280
	ACT V	300
The Comedy of Errors	PROLOGUE	320
	ACT II	340
	ACT III	360
	ACT IV	380
	ACT V	400
The Tragedy of Coriolanus	ACT I	420
	ACT II	440
	ACT III	460
	ACT IV	480
	ACT V	500
Cymbeline	ACT I	520

Zoom: 50

Other laws of movement

Generalizing Fitts' law to 2D pointing

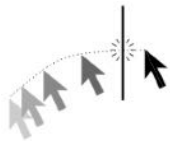


$$ID_{W'} = \log_2 \left(\frac{D}{W'} + 1 \right)$$

$$ID_{min} = \log_2 \left(\frac{D}{\min(W, H)} + 1 \right)$$

$$ID_{az} = \log_2 \left(\left[\omega \left(\frac{D}{W} \right)^p + \eta \left(\frac{D}{H} \right)^p \right]^{\frac{1}{p}} + 1 \right)$$

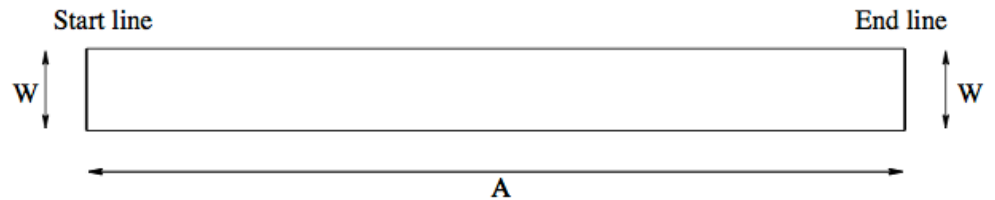
Goal-passing / crossing (Accot & Zhai)



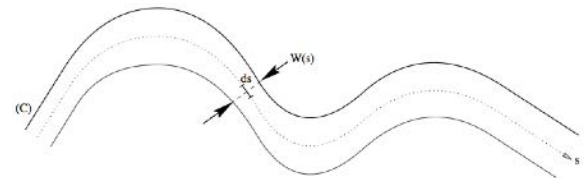
$$ID = \log_2 \left(\frac{D}{W} + 1 \right)$$

Steering law (Accot)

Tunnel: $MT = a + b \frac{A}{W}$



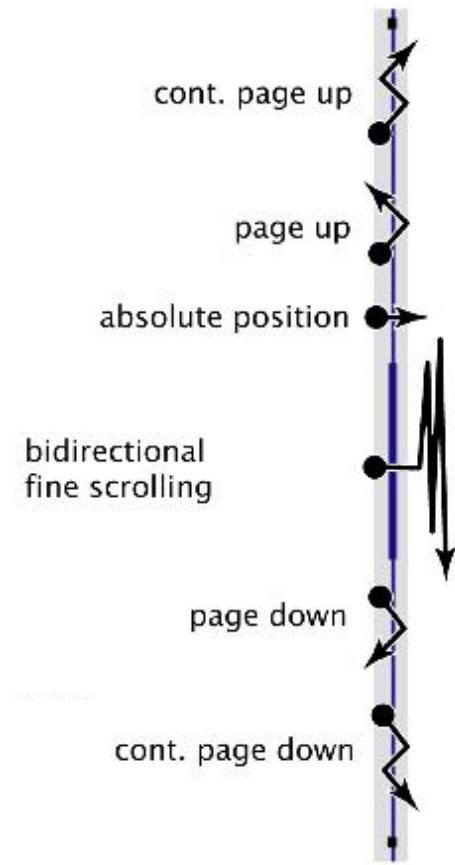
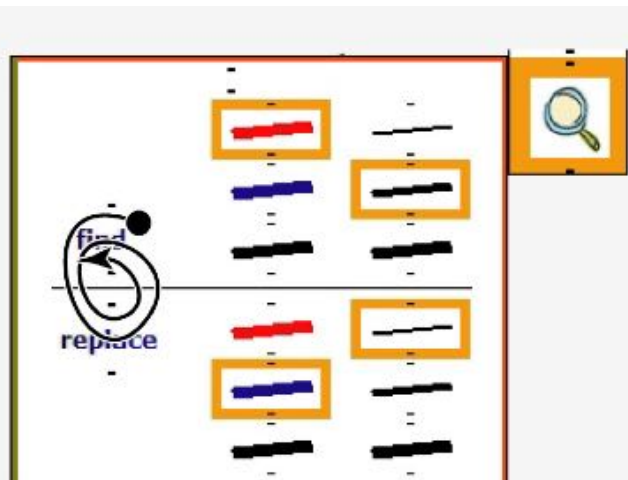
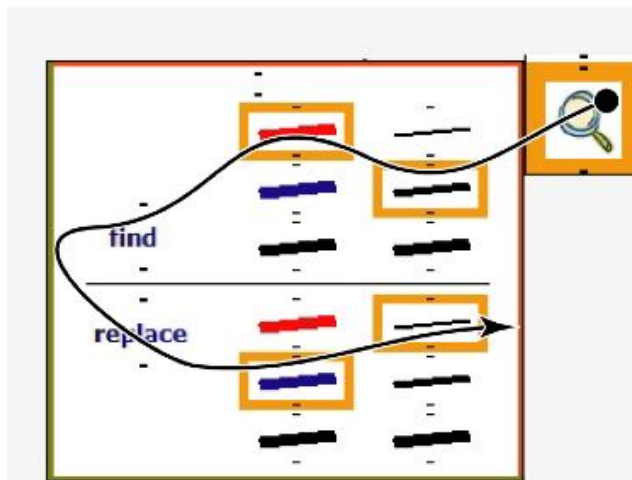
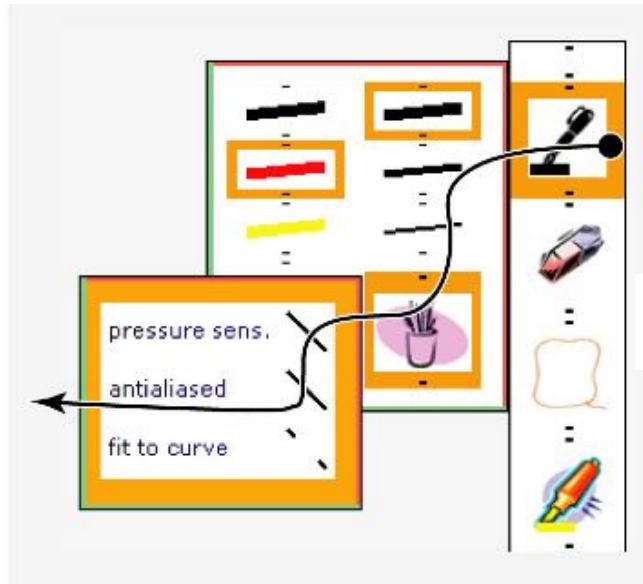
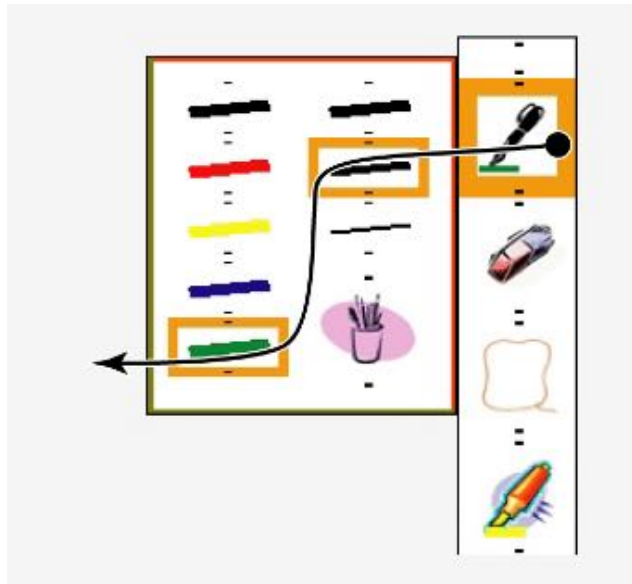
General case: $T_c = a + b \int_c \frac{ds}{W(s)}$

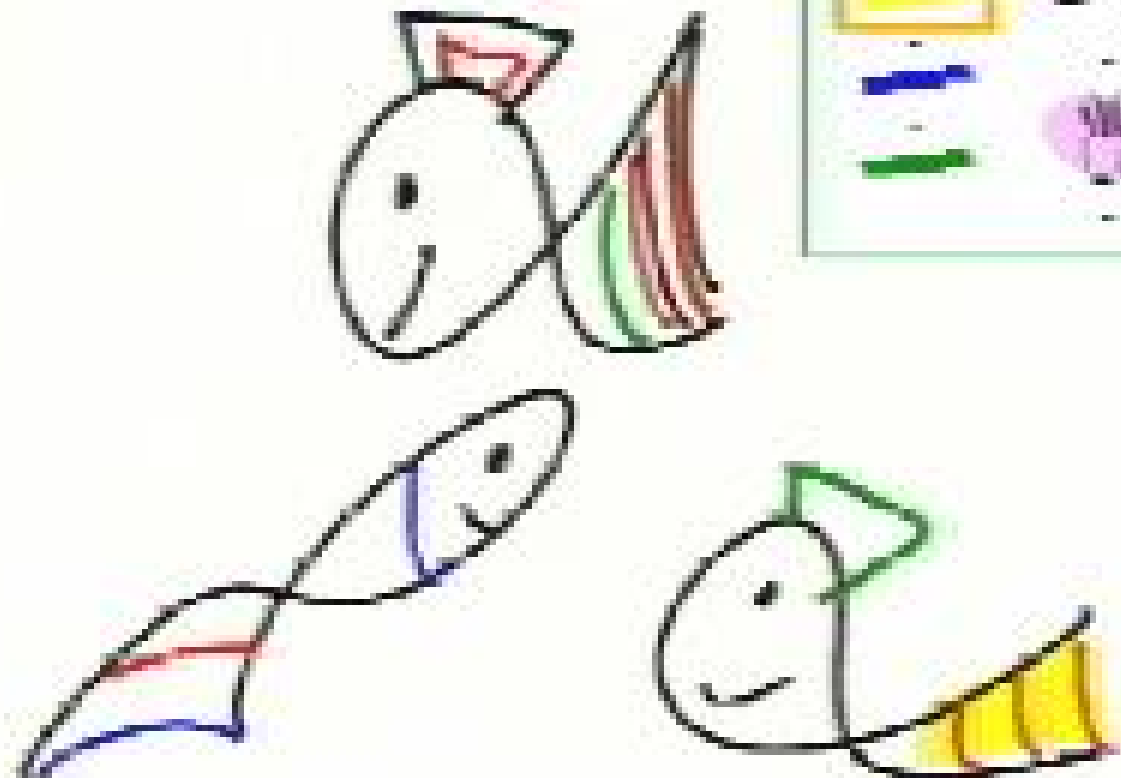
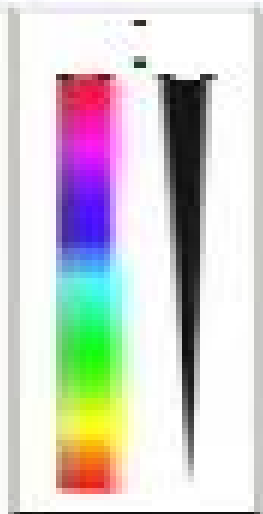


Crossy - crossing-based interface (Aplitz & Guimbretiere)



Crossy - crossing-based interface (Aplitz & Guimbretiere)





Crossy - Apitz & Guimbretière, 2004

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