Input

Input devices

Text entry

Positional input

MacBook Wheel (The Onion)

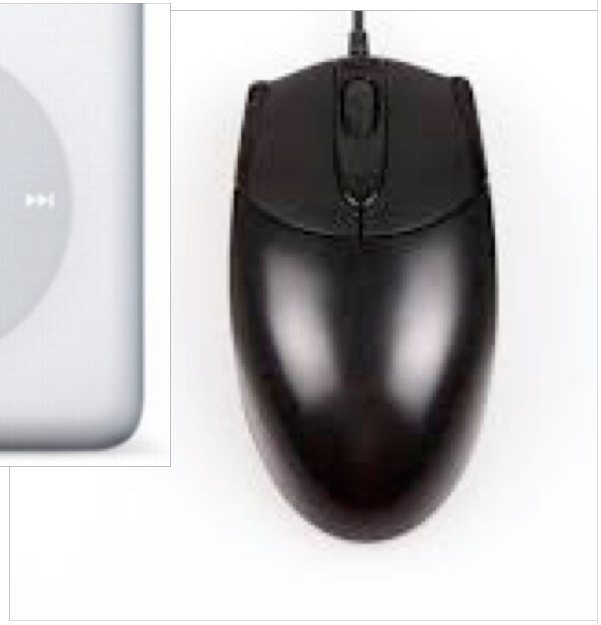
- https://youtu.be/9BnLbv6QYcA
- https://www.student.cs.uwaterloo.ca/~cs349/videos/macbook_wheel.mp4

iPod Wheel


iPod
Music
Photos
Extras
Settings
Shuffle Songs
Backlight
MENU

Dimensions to Classify Computer Input

- Sensing Method
- Continuous vs. Discrete
- Degrees of Freedom (DOF)
- Type of Data Managed



Specific vs. General Input Devices

- Specific input devices optimized for specific tasks
  - Problems?
- General input devices adapted to many task
  - Problems?

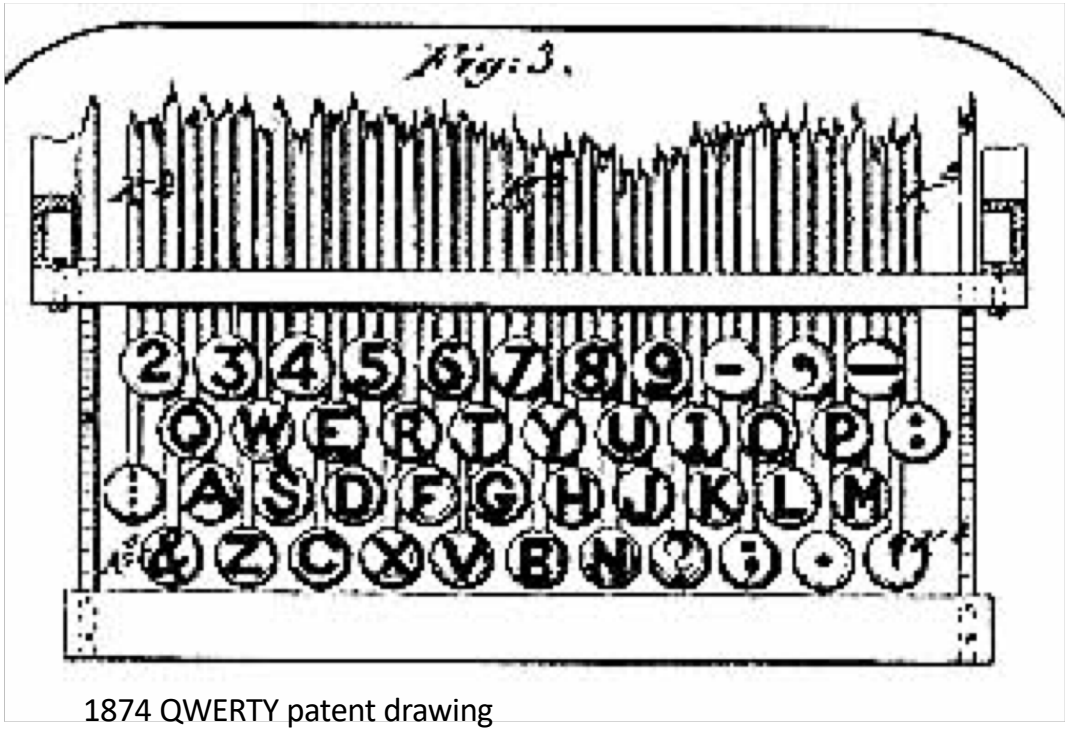
Text Input

QWERTY

Physical vs. virtual keyboards

Typewriters and QWERTY

- Original design intended for typing on paper
- QWERTY not designed to slow typing down → designed to space “typebars” to reduce jams and speed typing up

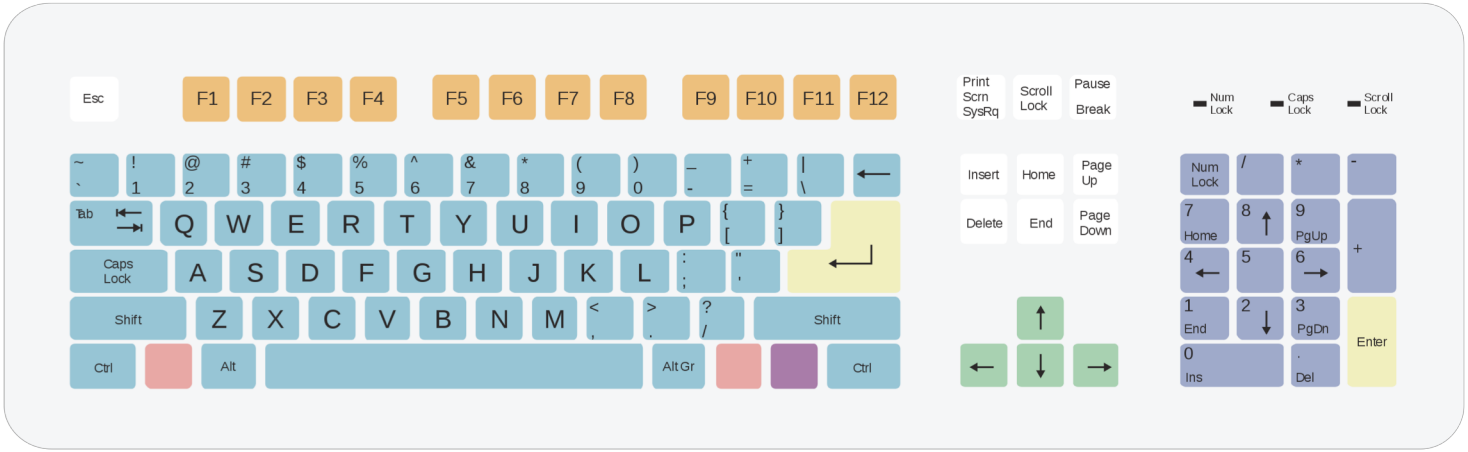


1874 QWERTY patent drawing

http://www.daskeyboard.com

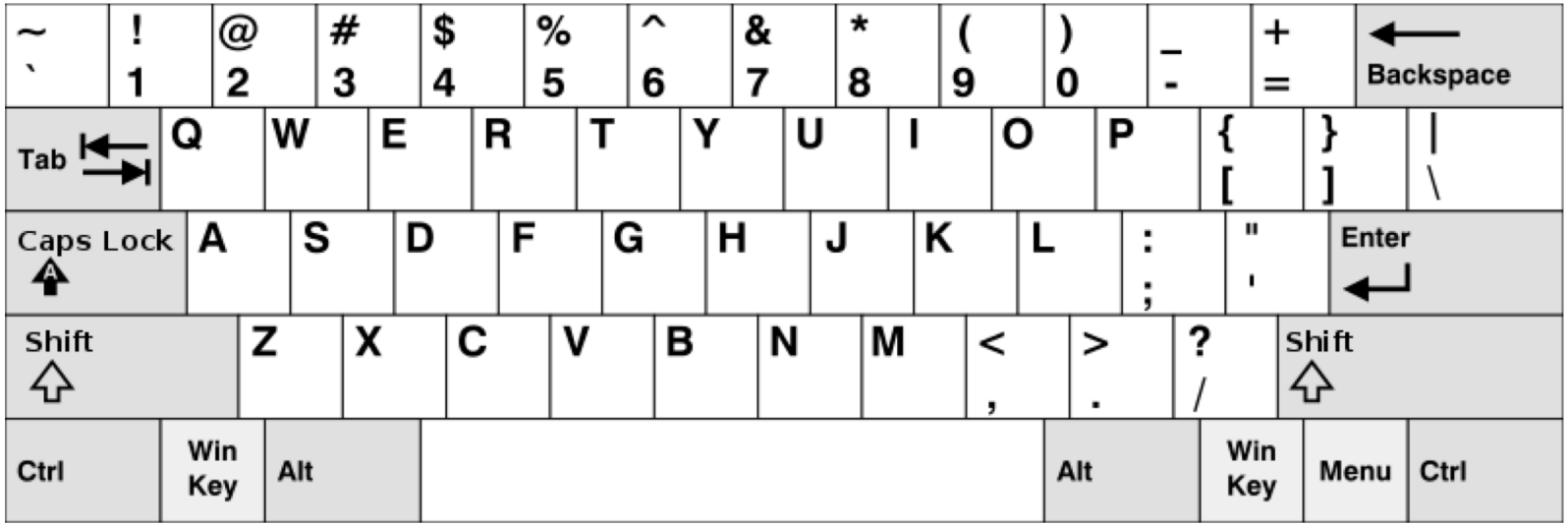
QWERTY Properties

- Standard for Latin-script alphabets
- Properties
  - Alternate hands when typing, for improved efficiency
  - Computer version adds function keys, cursor keys, meta keys
  - Can be modified for different locales (e.g. pound key, accents)
  - Variants also exist: AZERTY (French), QWERTZ (Czech)



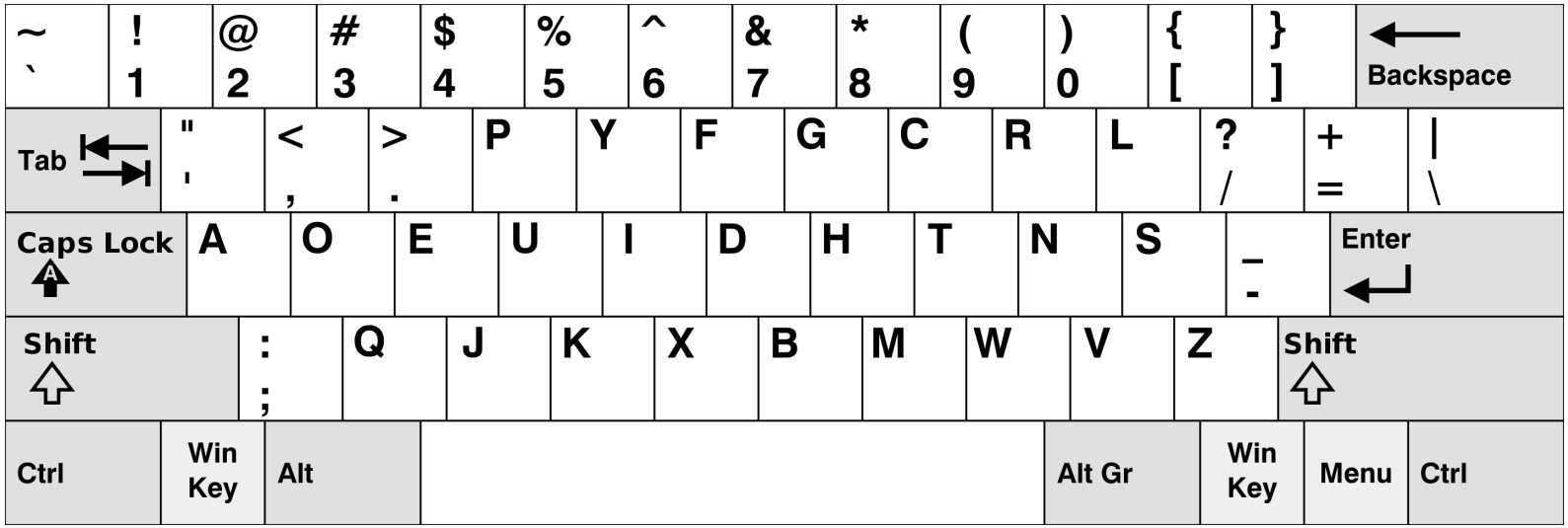
QWERTY Problems?

- Common combinations
  - awkward finger motions. (eg: tr)
  - a jump over home row. (eg: br)
  - are typed with one hand. (e.g. was, were)
- Most typing with the left hand (thousands of words left vs. hundreds right)
- About 16% of typing uses lower row, 52% top row, 32% home row



Dvorak Optimizations

- The most common letters and digraphs should be the easiest to type. Thus, about 70% of keyboard strokes are on home row.
- The least common letters should be on the bottom row, which is the hardest row to reach.
- The right hand should do more of the typing, because most people are right-handed.
- Is it actually more efficient? If so, at what cost?



How do keyboards work?

- Pressing a key generates a key code i.e. unique numeric value passed to app.

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Unicode

- The world needs more than 255 characters!
  - (Extended) ASCII was limited to 255 characters (i.e. 8 bits).

- **Unicode** is a superset of ASCII, that has replaced it in common use
  - Values 0-127 have the same meaning in both (e.g. ‘A’ == 65)
  - Uses multiple bytes to store character information, which greatly increases the range of values
  - Denoted as UTF-xx where xx is the minimum number of bits.

- UTF-8 is the standard method of encoding characters
  - minimum 8 bits.
  - Capable of encoding all 1,112,064 code points in Unicode (characters, control codes, other meaningful characters)

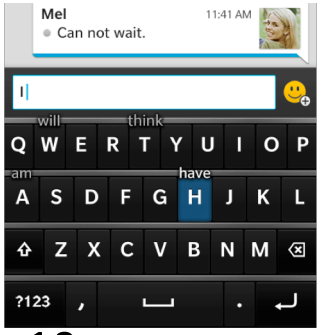
Mechanical Design of Keyboards

- To increase portability of devices, keyboards are frequently downsized
  - Smaller, low-profile keys
  - Shorter travel distance
  - Sometimes fewer keys
- All interfere with typing, or reduce efficiency!

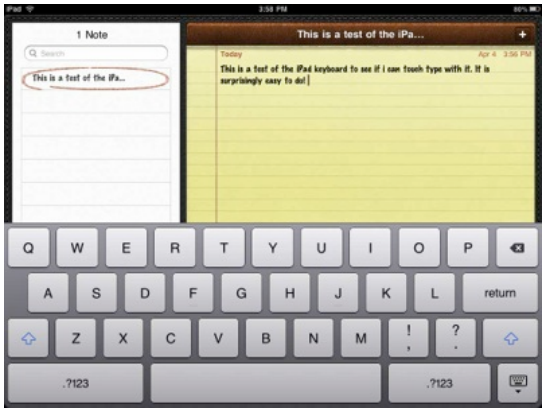
“Virtual” Keyboards

- Touch screen or other flat surface
- Problems:
  - small keys reduce accuracy
  - no mechanical feedback makes it hard to tell if key was pressed
  - no tactile feedback makes it hard to find the home row
  - resting of hands difficult
- Advantage:
  - portable, no extra hardware
  - customizable keys (e.g. new language, symbols, emojis)
  - customizable layout or functionality (e.g. swipe, thumb layout)

iPhone



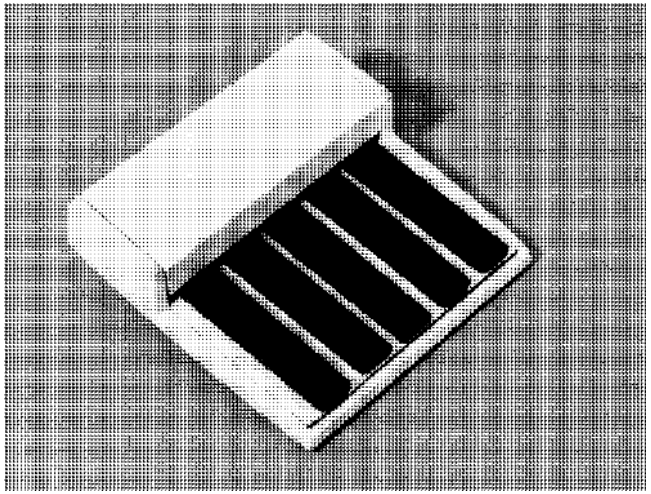
z10

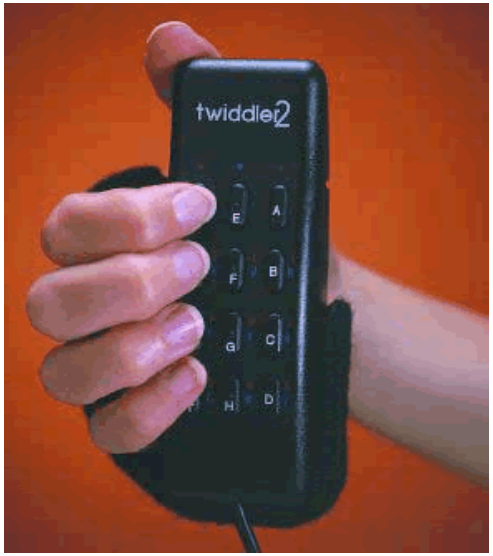


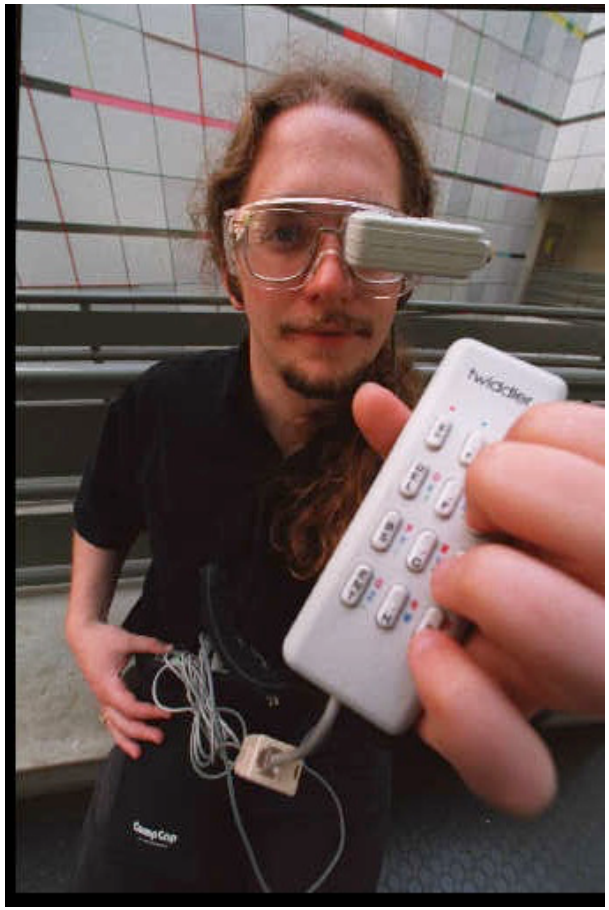
iPad

Chording Keyboards

- Englebart's NLS Keyboard
  - Multiple keys together produce letter
  - No hand "targeting", potentially very fast
  - Can be small and portable
  - One handed
- Thad Starner's Twiddler
  - for wearable computing input

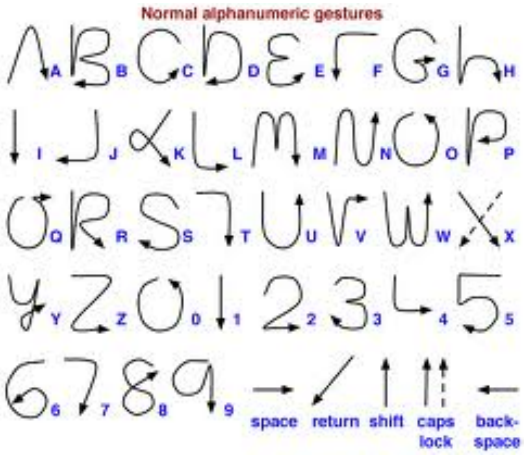


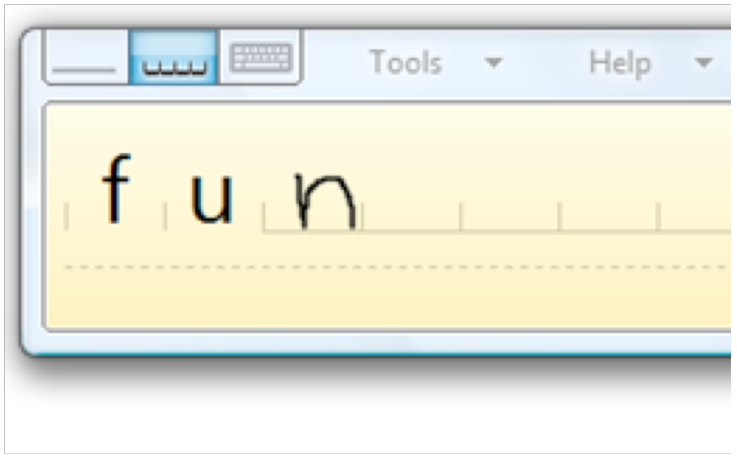




Text Recognition and Gestures

- Graffiti/Unistroke Gestures
  - map single strokes to “enter letter” commands
- Natural Handwriting recognition
  - dictionary-based classification algorithms

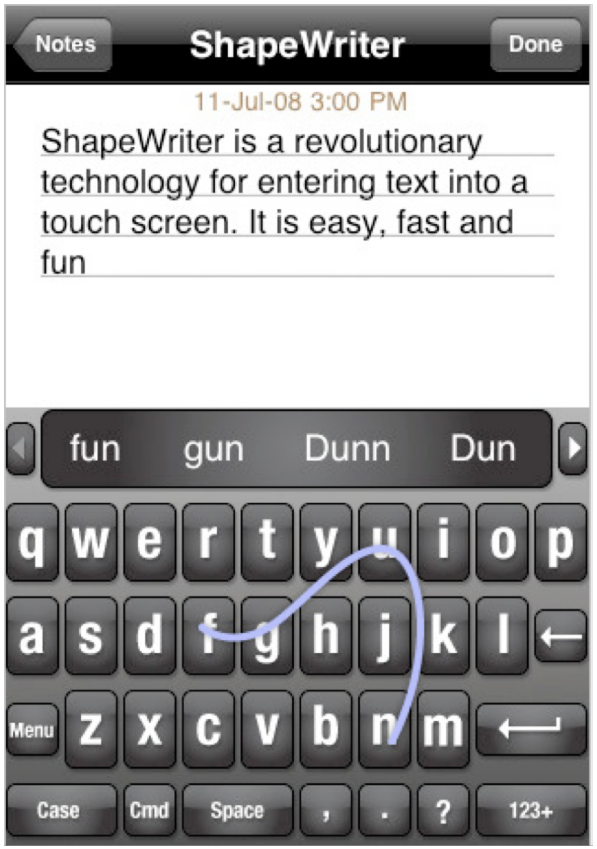




Predictive Text

- Use language characteristics to predict input
  - Given characters typed so far, what are the most likely intended words?
  - Given words typed so far, what is the most likely word to follow?
- A variation is used for T9, nine-key text entry
  - Given an ambiguous set of characters, what is the most likely word
- Possible Problems
  - “collisions” between common words
  - entering words not in dictionary difficult
  - hard to focus on typing and monitoring prediction

Gestural Text Input



ShapeWriter

http://www.shuminzhai.com/shapewriter_research.htm

8Pen Keyboard

http://www.8pen.com/

Text Input Expert-User Input Rates

Device	Input Rates
Qwerty Desktop	80+ WPM typical, record: 150 WPM for 50 minutes
Qwerty Thumb	60 WPM typical with training (Clarkson et al., CHI 2005)
Soft Keyboards	45 WPM
T9	45 WPM possible for experts (Silverberg et al., CHI 2000)
Gestural	~30 WPM 8Pen, ShapeWriter claims 80 WPM (expert)
Handwriting	33 WPM (Wilkund et al., Human Factors Society, 1987)
Graffiti 2	9 WPM (Koltringer, Grechenig, CHI 2004)

Positional Input

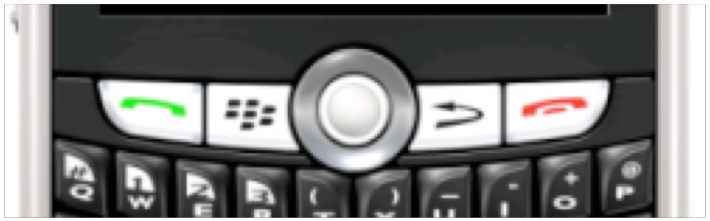
Positional devices

Direct vs. indirect

Absolute vs. relative

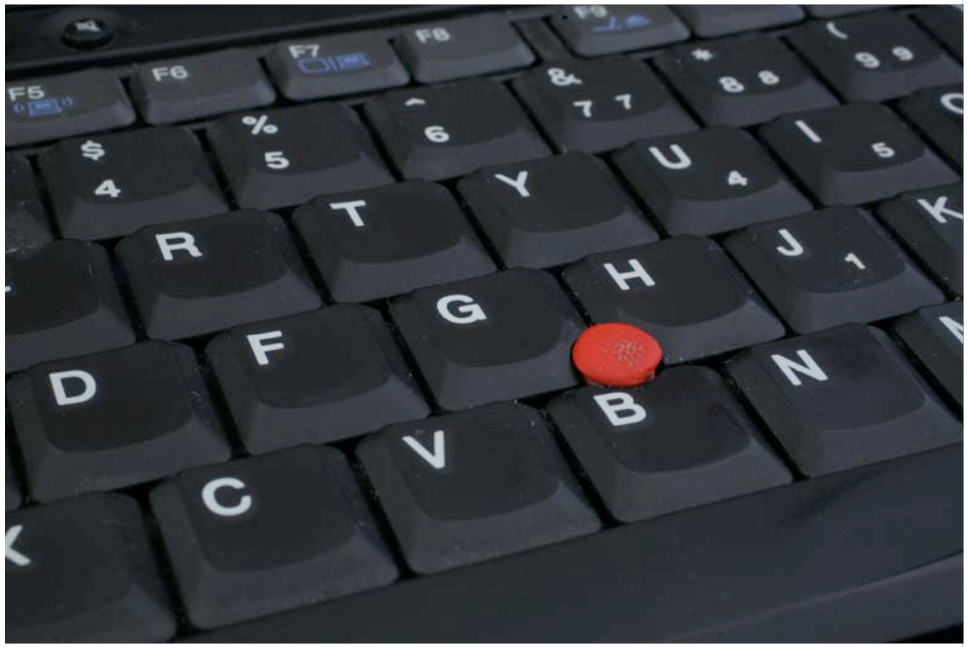
Properties of Positional Input Devices

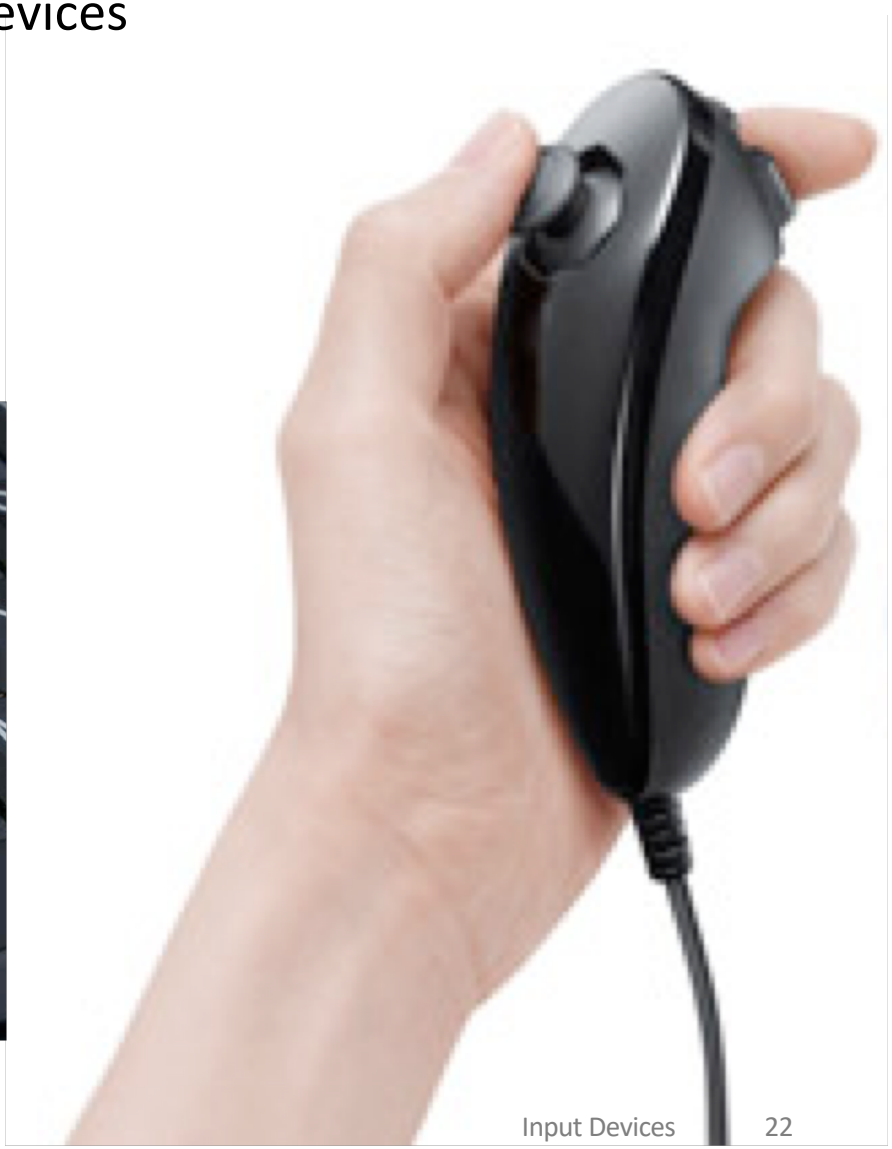
- Force vs. Displacement Sensing
  - (most) joysticks = force
  - mouse = displacement
- Position vs. Rate Control
  - (most) joysticks = rate
  - mouse = position
- Absolute vs. Relative Positioning
  - touchscreen = absolute
  - mouse = relative
- Direct vs. Indirect Contact
  - direct = touchscreen
  - indirect = mouse
- DOF (Dimensions) Sensed



Force vs. Displacement Sensing

- Force == Isometric
  - elastic isometric vs. “pure” isometric devices
  - e.g. joysticks
- Displacement == Isotonic
  - e.g. mouse

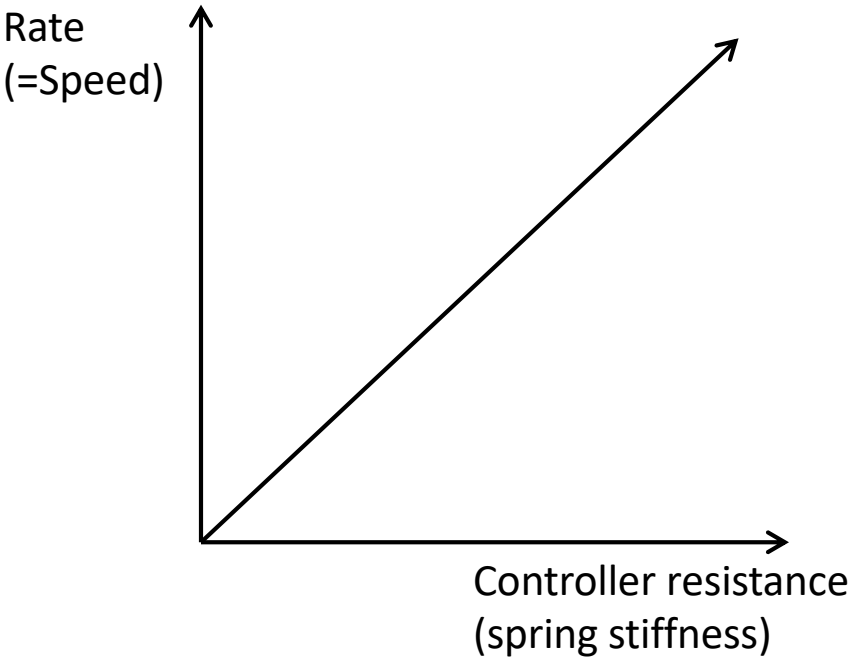




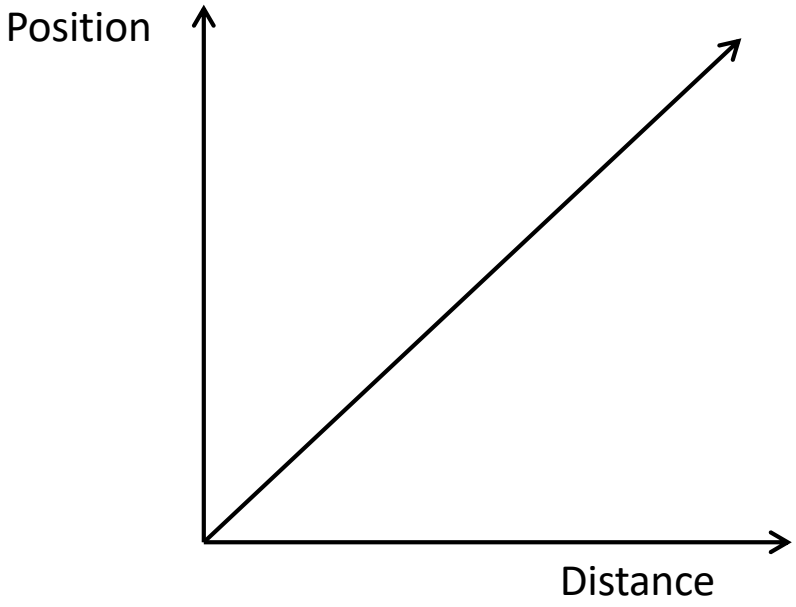
Position vs. Rate Control Transfer Function

- force sensing should be mapped to rate (e.g. joystick, pedal)
- displacement sensing should be mapped to position (e.g. mouse)

Force sensing
(rate control)

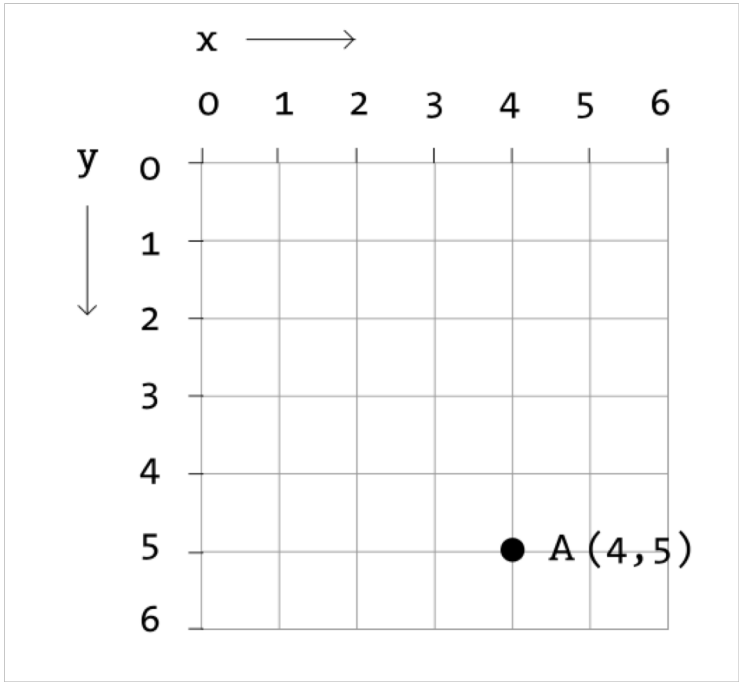


Displacement sensing
(position control)



Position Control & Managing Coordinates

- We'll discuss this more when we discuss drawing (later!).
- Cartesian coordinates vs. computer coordinates



Positional control (displacement sensing) devices are more common for desktop interaction, and report screen coordinates.

- Clicking: point
- Dragging: series of points

Direct vs. Indirect Input

- Indirect: the position of the cursor is controlled by some external device.

- Direct: the position of the cursor is controlled by direct contact with the screen.

Absolute vs. Relative Position

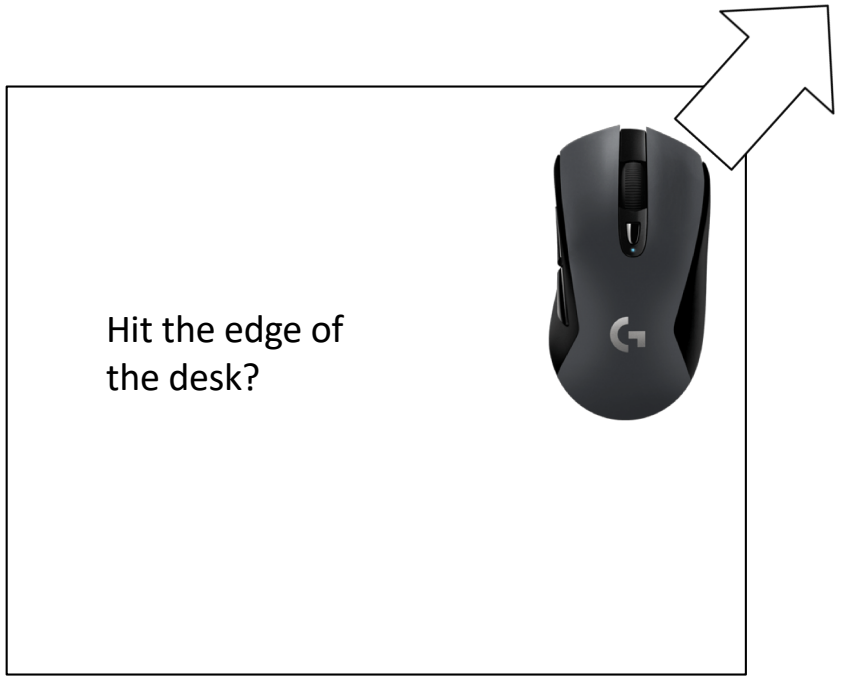
- Absolute position is a direct mapping of input device position to a display input position. Examples?

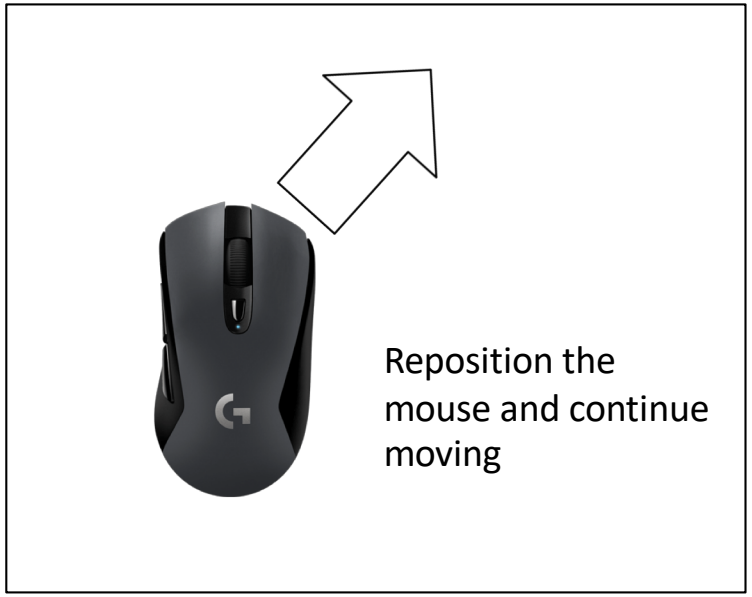

- Relative position maps changes in input device position to changes in display input position. Examples?



Clutching

- **Scenario**: you're moving the mouse and you hit the edge of the desk before you finish positioning the mouse. What do you do?
- To make relative position work, you need to **clutch** (i.e. repeatedly move to achieve the target)
  - clutching is <u>one</u> solution to making relative motion work

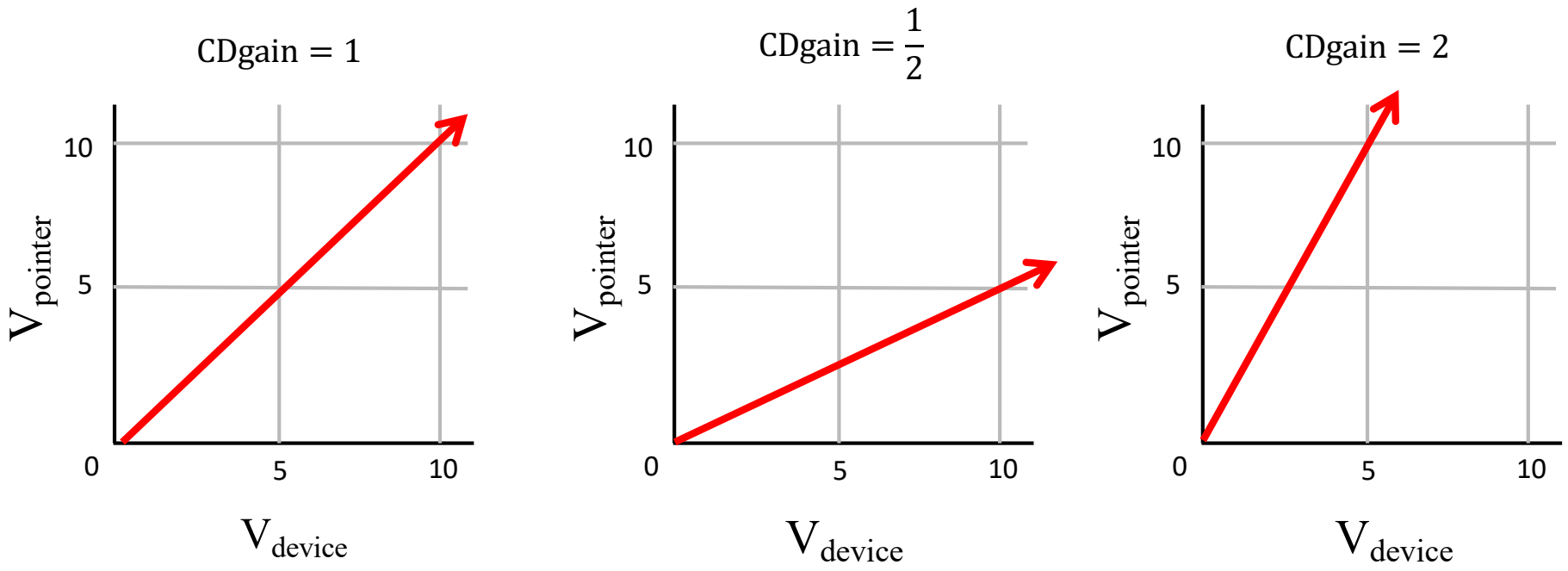




Control-Display Gain (CD Gain)

- Ratio of ***display*** **pointer** movement to **device *control*** movement
  - the ratio is a scale factor (the "gain")
  - usually expressed in terms of velocity
  - works for rate control and position control

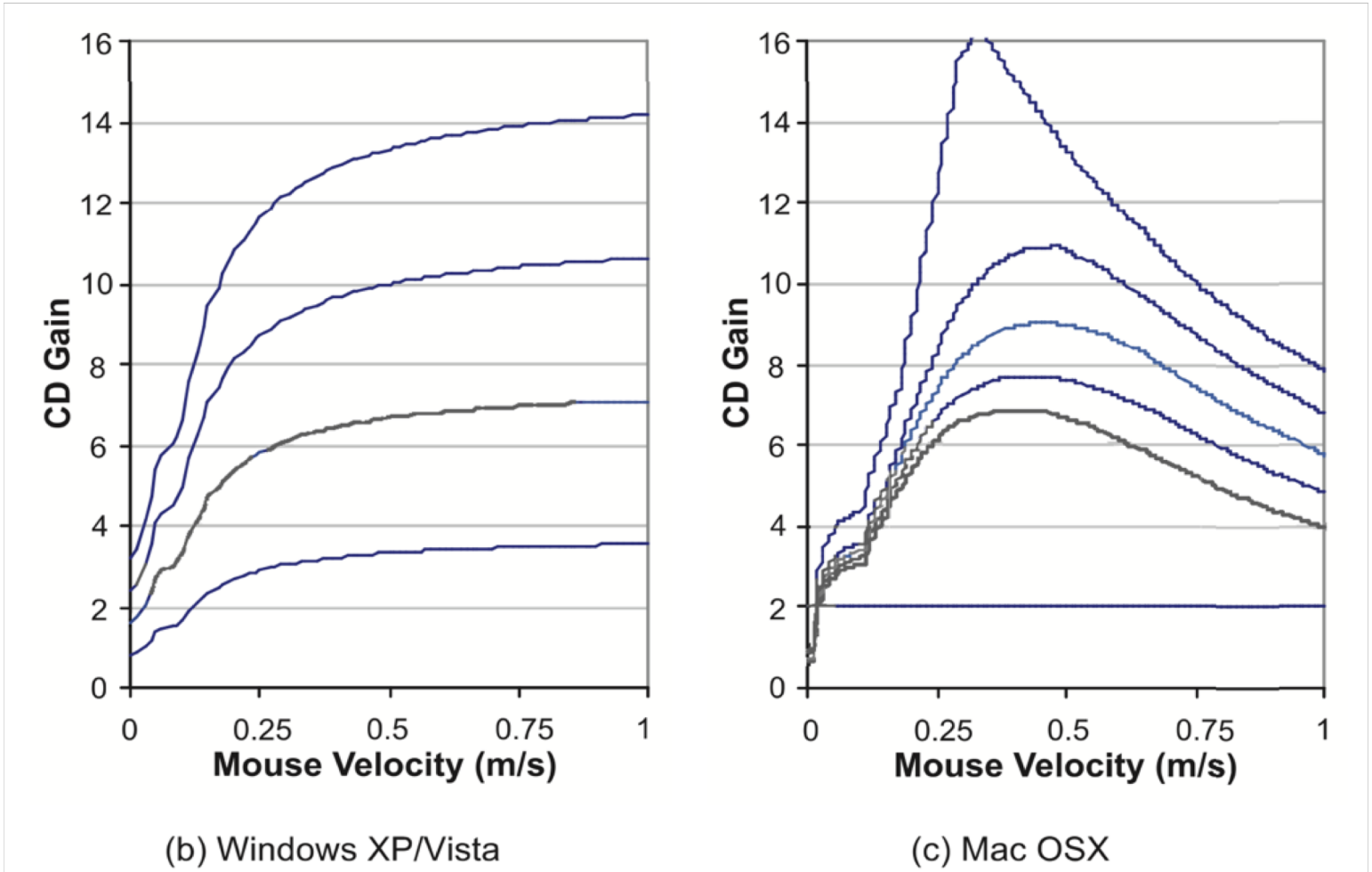
$$\text{CDgain} = V_{\text{pointer}} / V_{\text{device}}$$



Pointer Acceleration

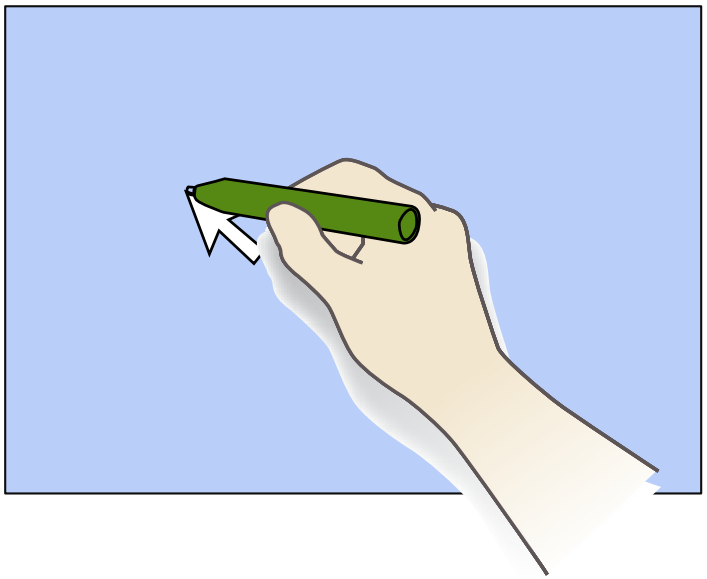
- Dynamically change CD Gain based on device velocity; can reduced the need to clutch

(Casiez et al. 2008)

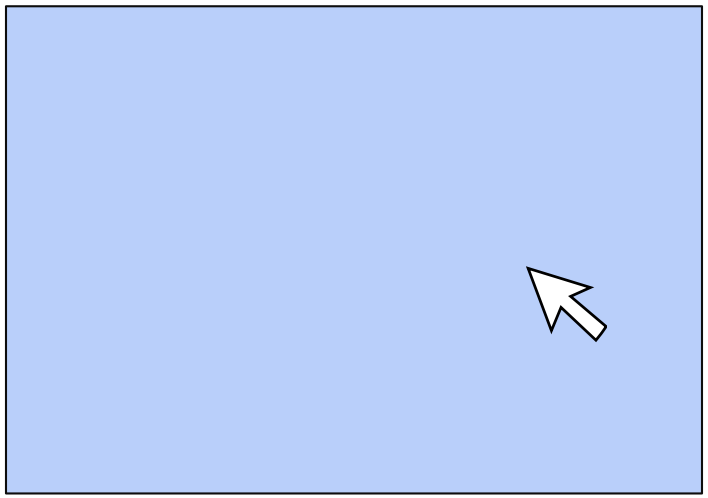


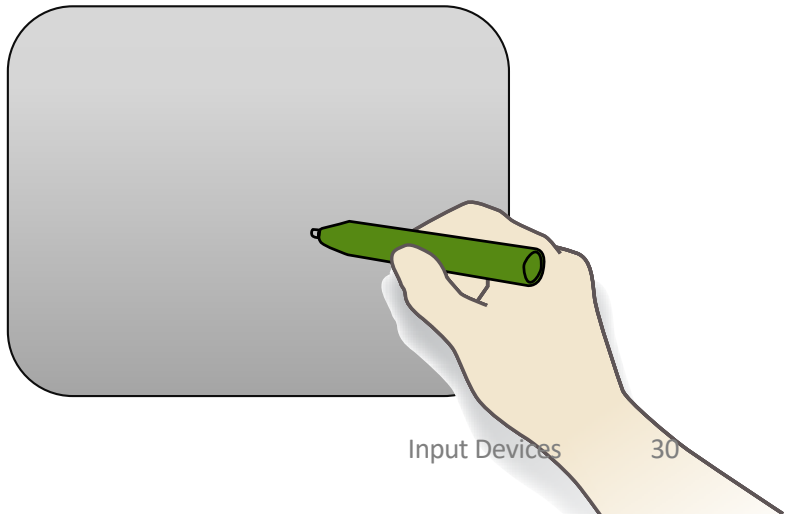
Combinations

Absolute Direct



Relative Indirect





Hybrid Absolute and Relative Pointing

- http://youtu.be/FZmOBIg5KjM