

AI and HCI: Towards Human-Computer Partnerships

Michel Beaudouin-Lafon
with thanks to Wendy Mackay

DatalA-JST - 11 July 2018

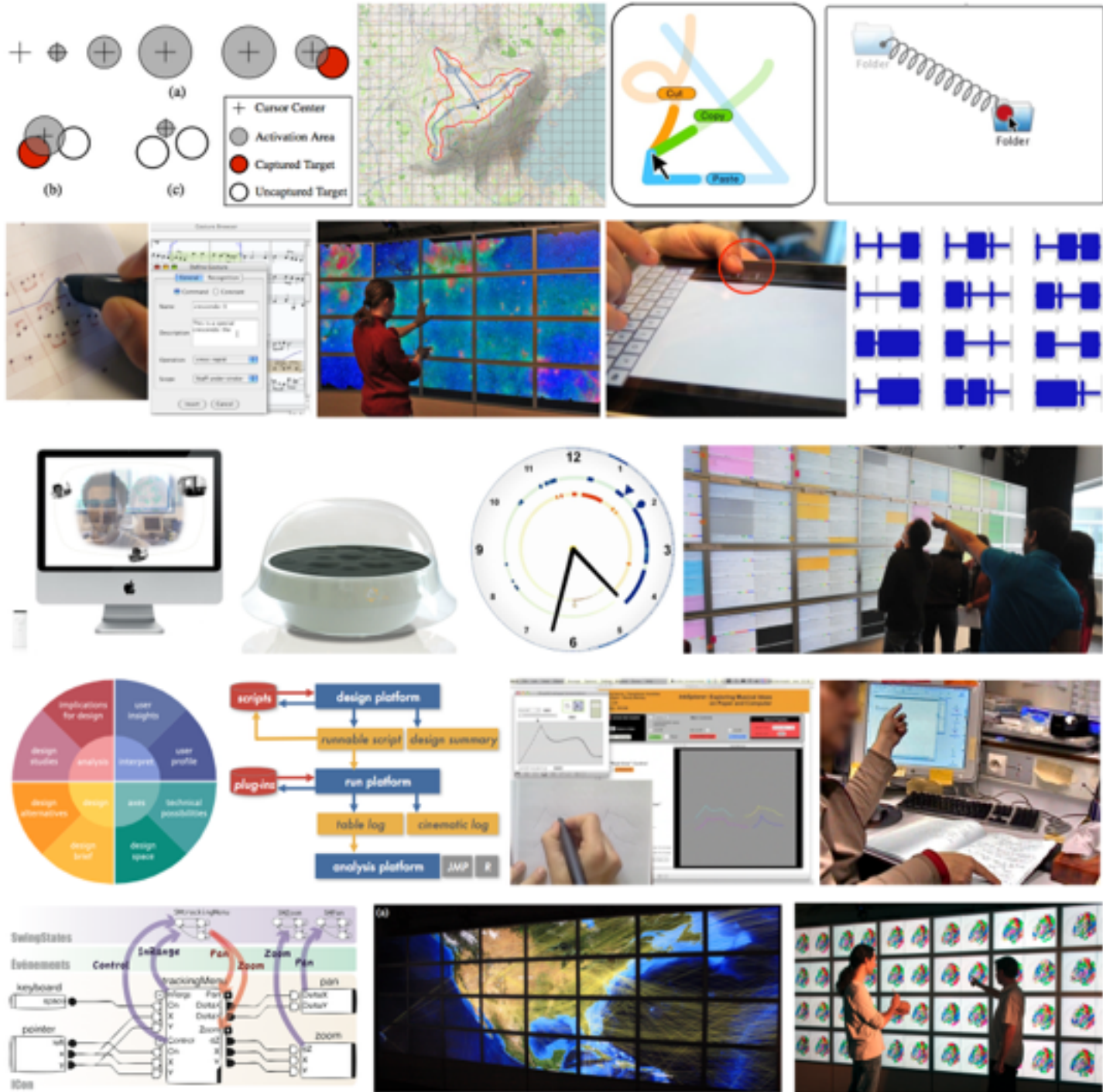


ex)situ

Extreme
Situated
Interaction

Human-
Centered
Computing

Partnerships
Creativity
Collaboration



How do we interact with computers ?

Computer as *tool*
Empower users



Human-
Computer
Interaction

Computer as *servant*
Delegate tasks



Artificial
Intelligence

Computer as *medium*
Communicate



Mediated
Communication

How do we interact with computers ?

AI perspective on HCI:

Human-in-the-loop

Use human input to *improve the algorithm*

How do we interact with computers ?

AI perspective on HCI:

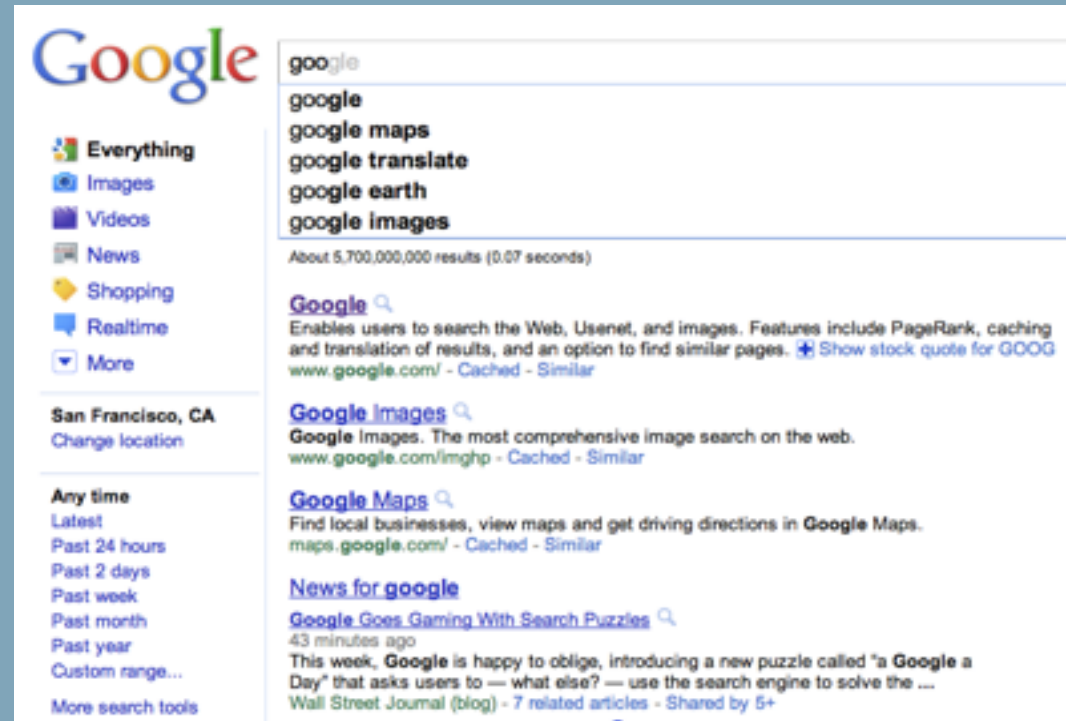
Human-in-the-loop

Use human input to *improve the algorithm*

User types

Google suggests

User chooses



But 'human-in-the-loop' is
not always best for the user



How do we interact with computers ?

If *Human-in-the-loop*
uses human input to *improve the algorithm*

How do we interact with computers ?

If *Human-in-the-loop*
uses human input to *improve the algorithm*

Shouldn't we also have
'*Computer-in-the-loop*' to *empower the user?*

Human-Computer Partnerships

We already create models of *users*
to inform the *system*

Shouldn't we also create models of the *system*
to inform the *user*?

We need to create effective
human-computer partnerships

What do we mean by 'partnership' ?

Drive a motorcycle
User in control



What do we mean by 'partnership' ?

Drive a motorcycle
User in control

Take a taxi
Driver in control



What do we mean by 'partnership' ?

Drive a motorcycle
User in control

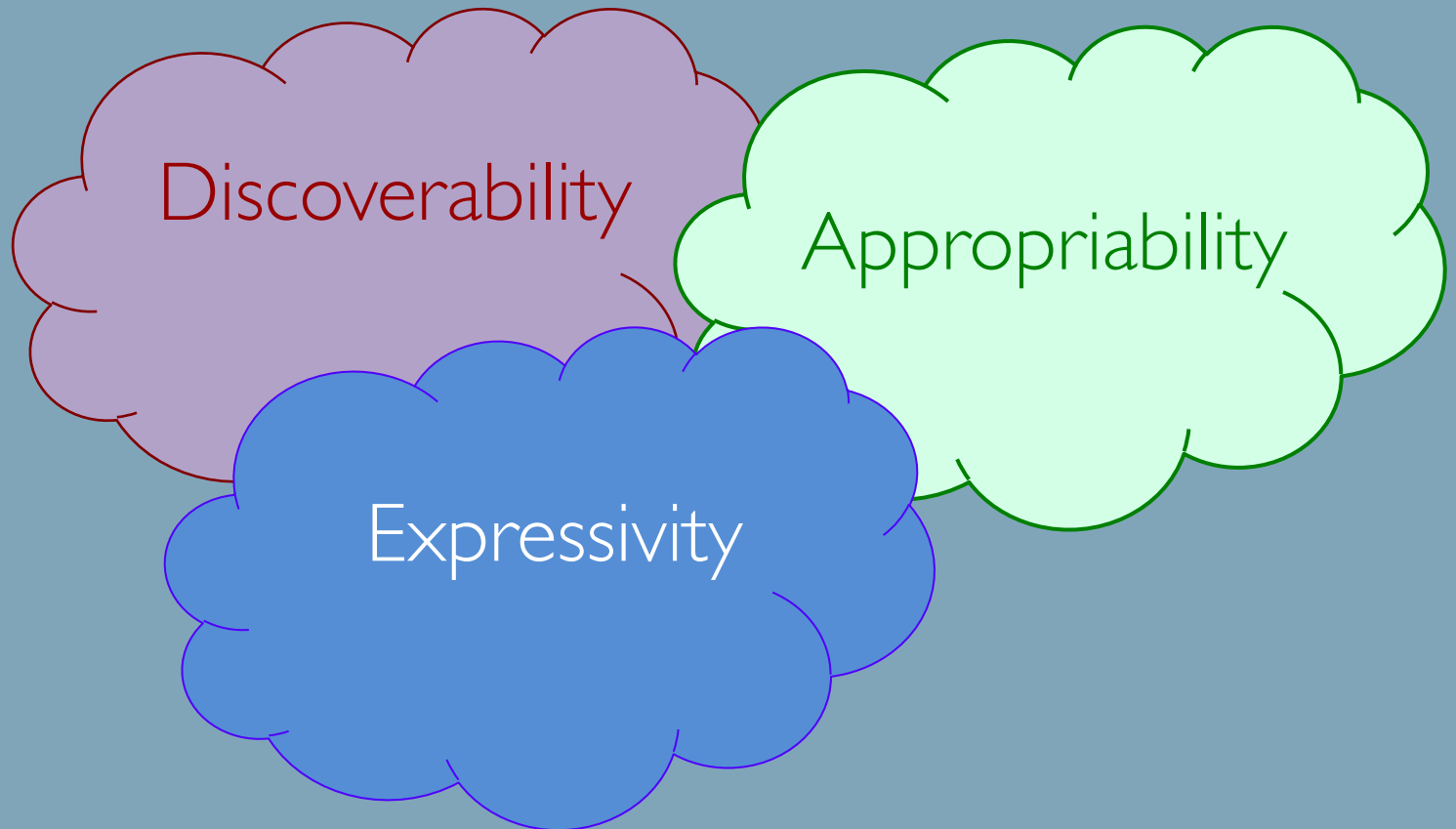
Take a taxi
Driver in control

Ride a horse
Shared control



Human-Computer Partnerships

To share control, users need:



What can we learn from physical tools ?

We can use physical tools as designed...



What can we learn from physical tools ?

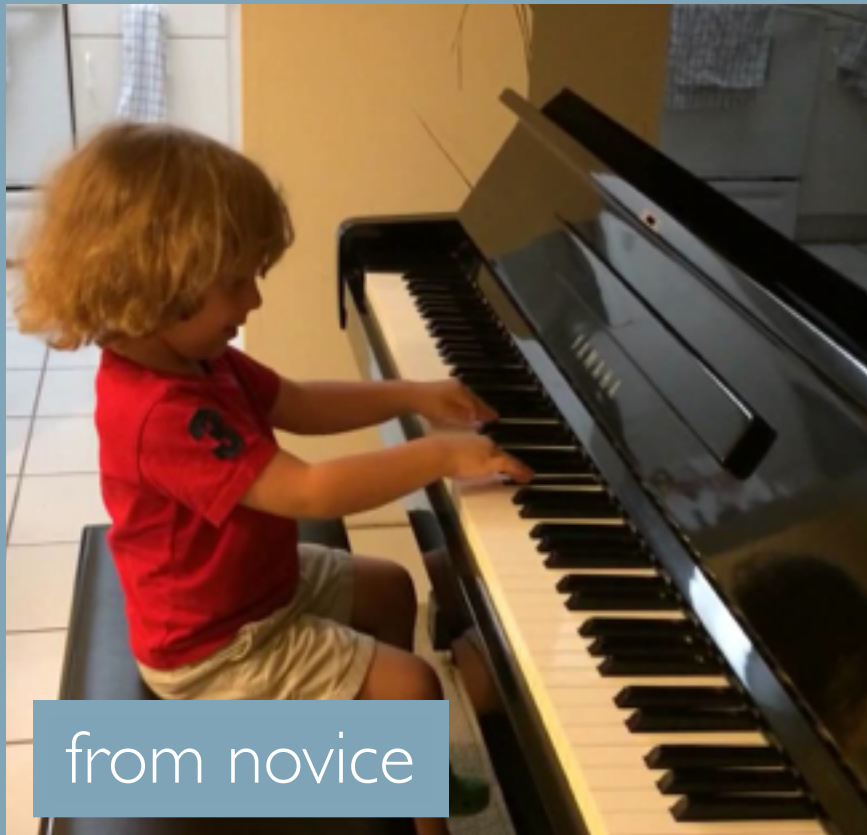
But we can also improvise !





Some of us become
virtuoso users

Why can't we learn to 'play' software tools ?
without relearning the interface
with every software upgrade ?



from novice

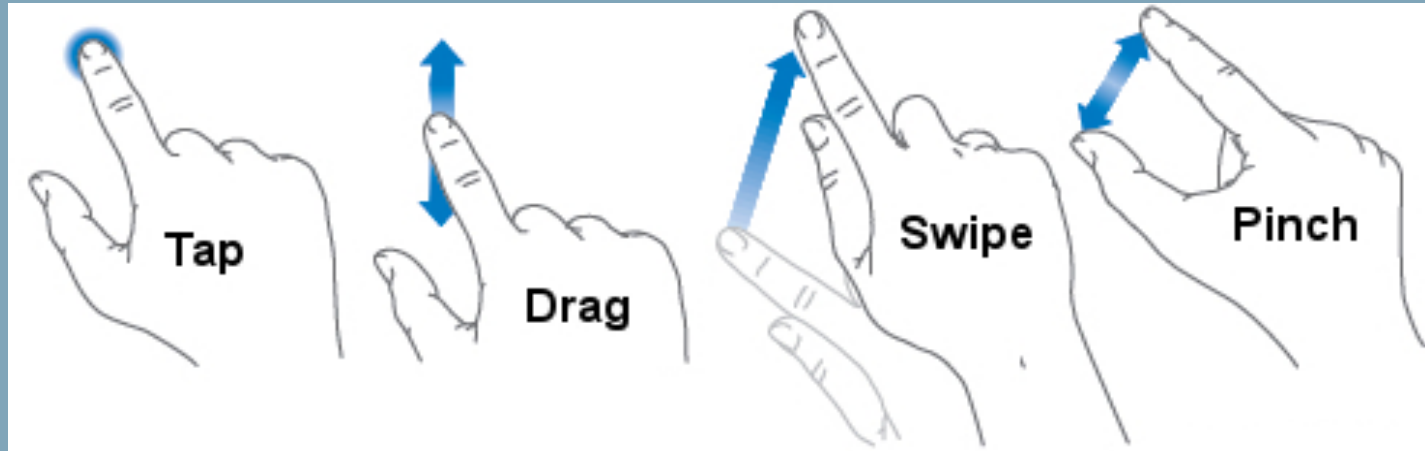


to expert

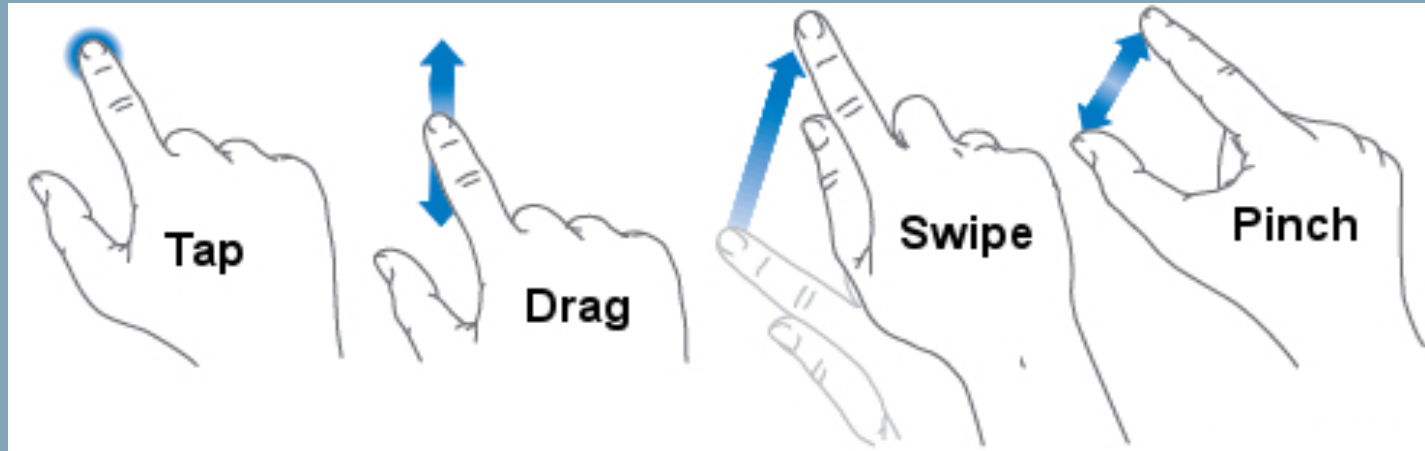
Take a smartphone ...



Smartphone interfaces are simple



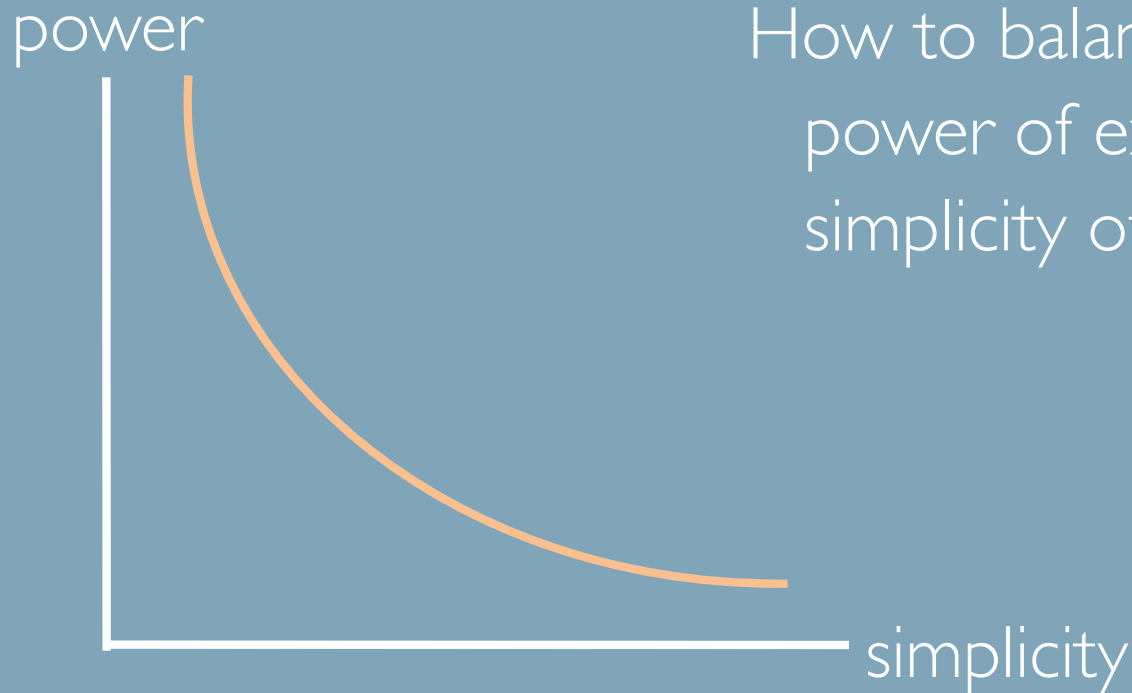
Smartphone interfaces are simple



Why not powerful, expressive *and* simple ?

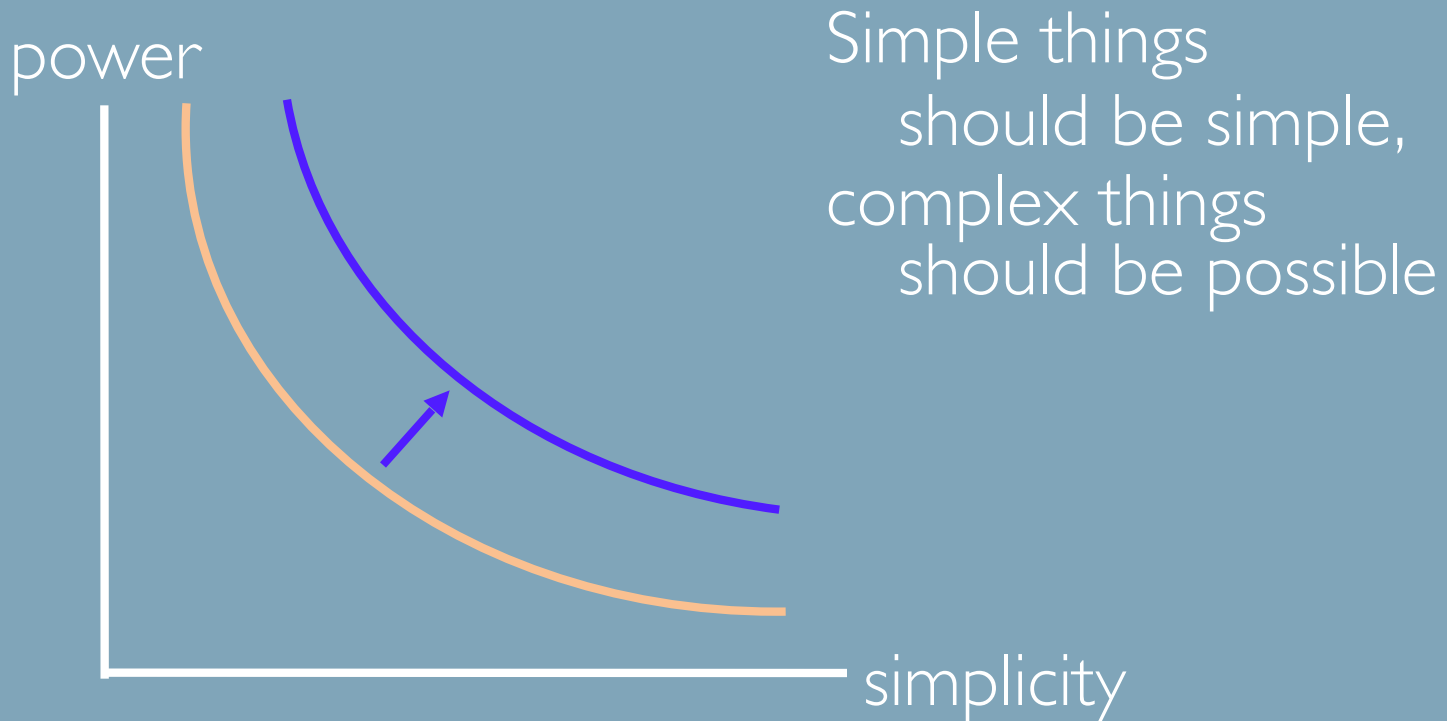


Major design trade-off



How to balance:
power of expression
simplicity of execution?

Solution: Shift the curve



Human-computer partnerships

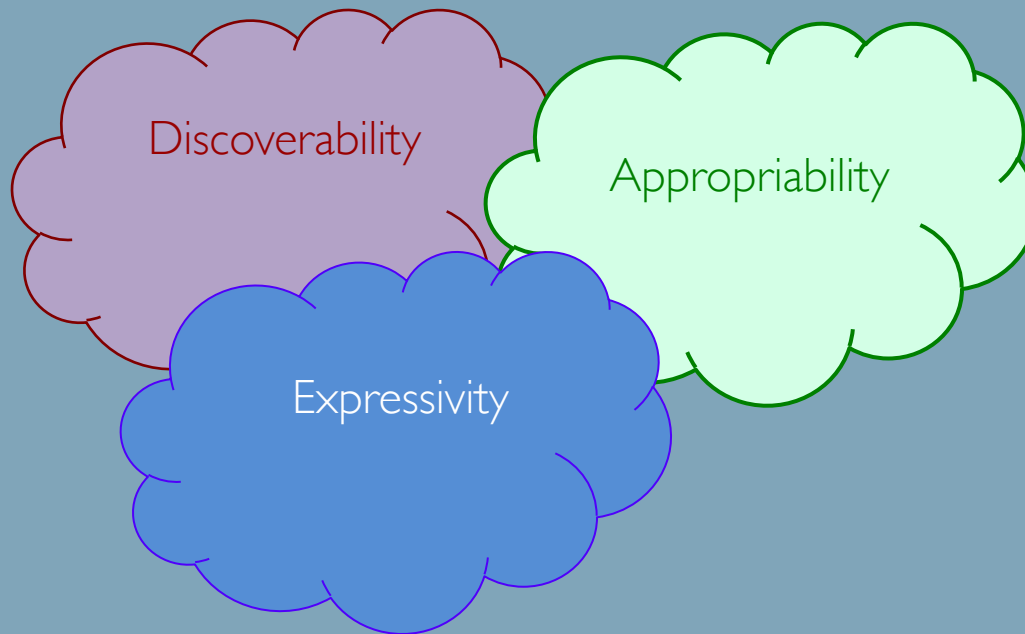
People can

adapt to technology

they learn it

adapt the technology

they appropriate it



Human-computer partnerships

People can

adapt to technology

they learn it

adapt the technology

they appropriate it

Computers can

adapt to people

they learn (AI)

adapt people's behavior

they teach (CAI)

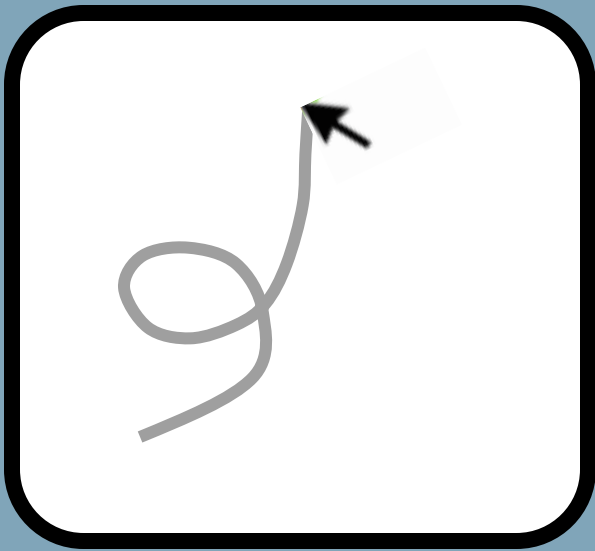


Discoverability

How can I learn
which gesture
executes which command?

Octopocus

Experts just perform the gesture

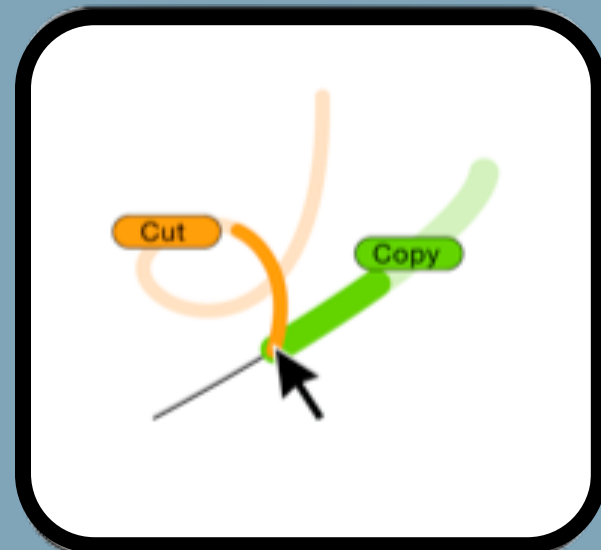
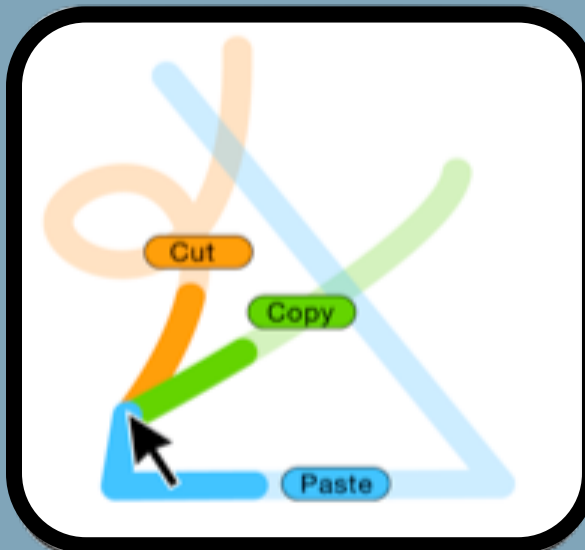
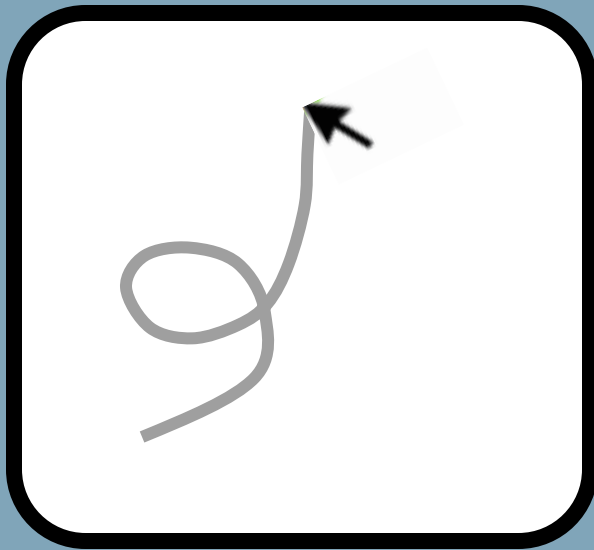


Octopocus

Experts just perform the gesture

Novices pause ...

and the Octopocus guide appears



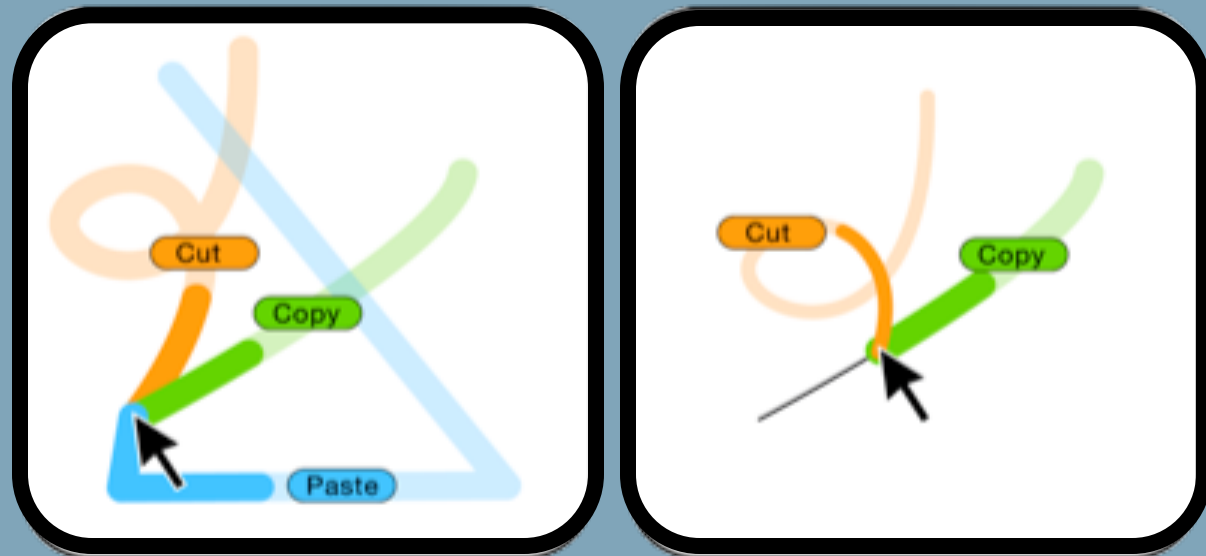
Octopocus

Progressive feedforward

What gestures are available ?

Progressive feedback

What did the system recognize ?



Inking the '*Help*' command



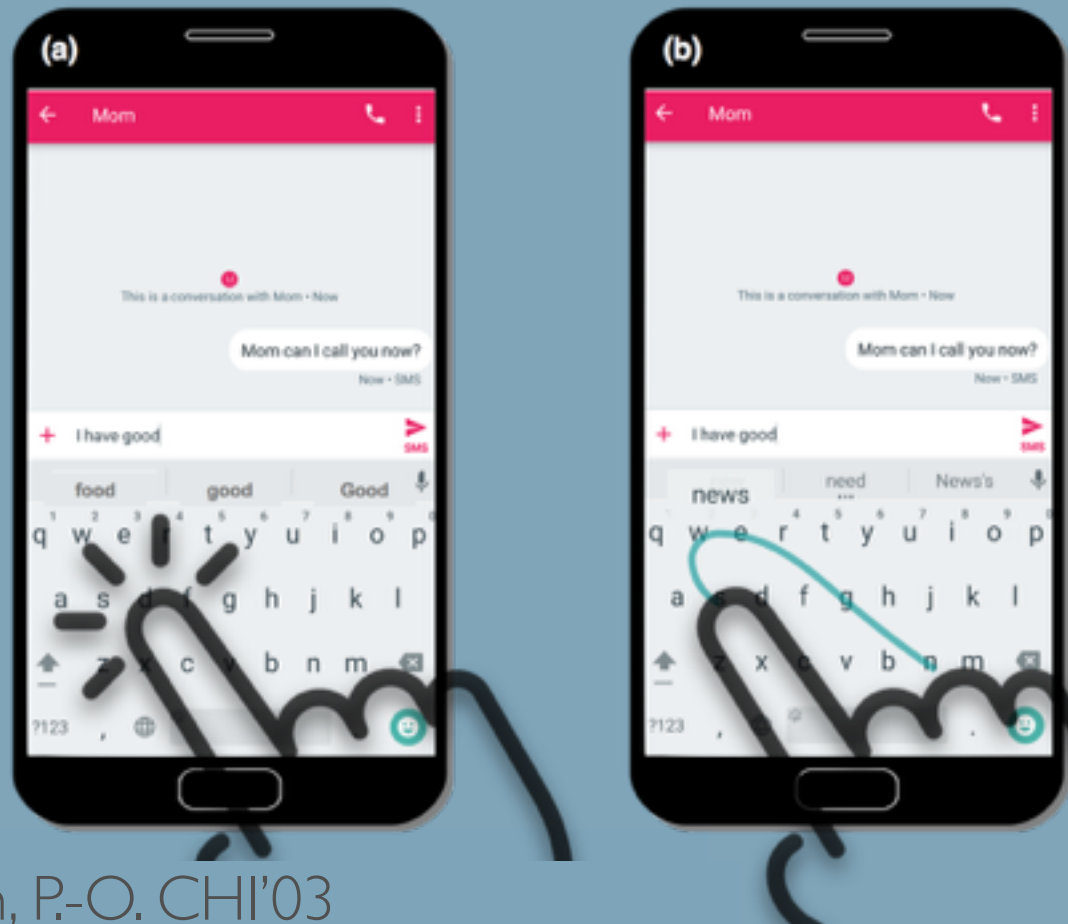
Discoverability

How can I learn
new gestures
that execute commands ?

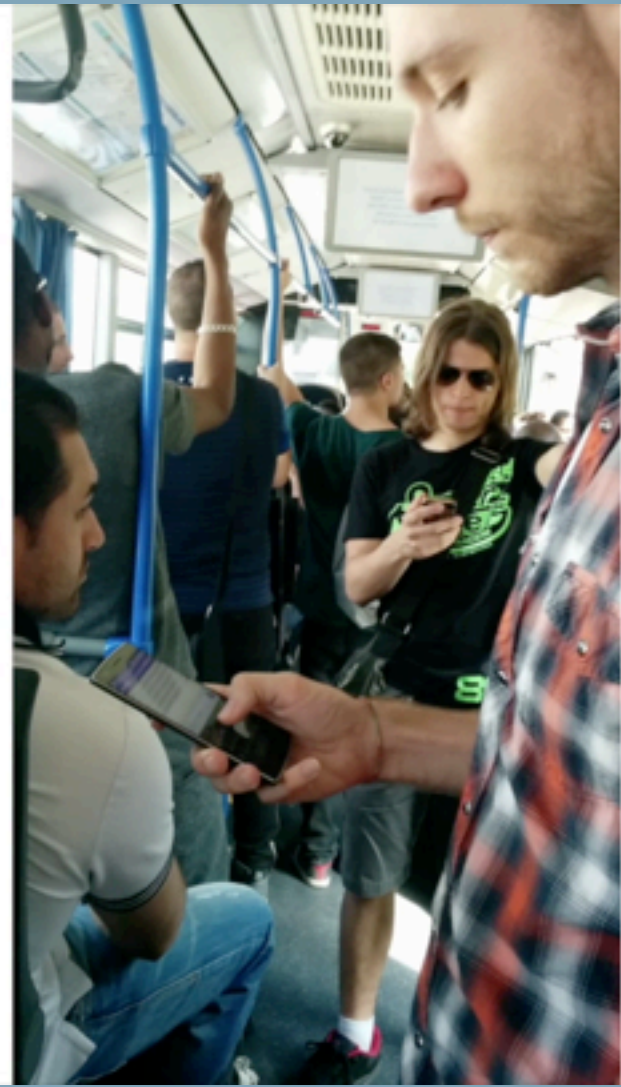
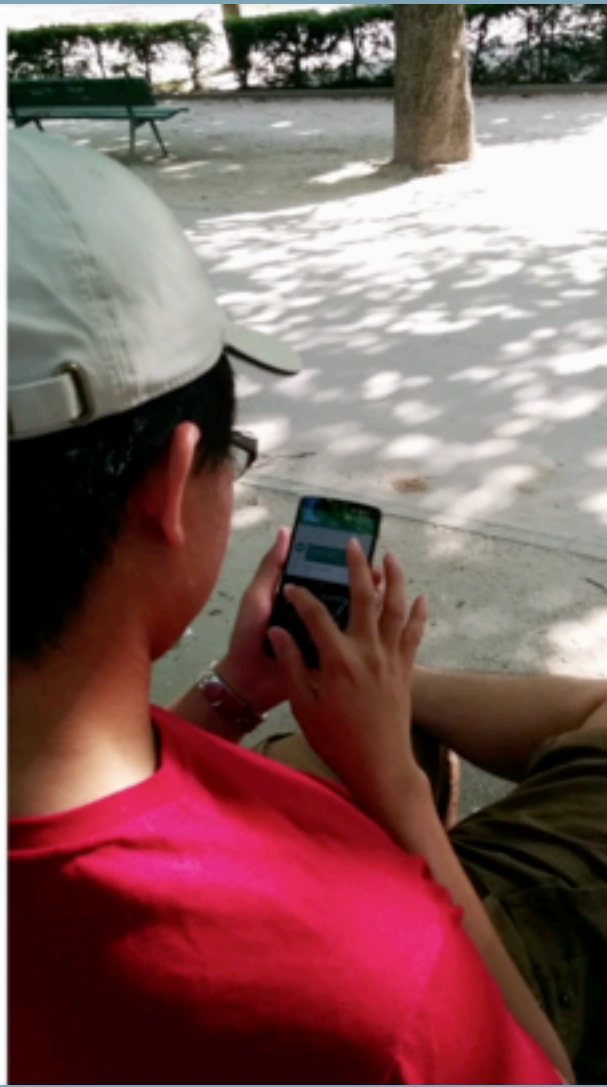
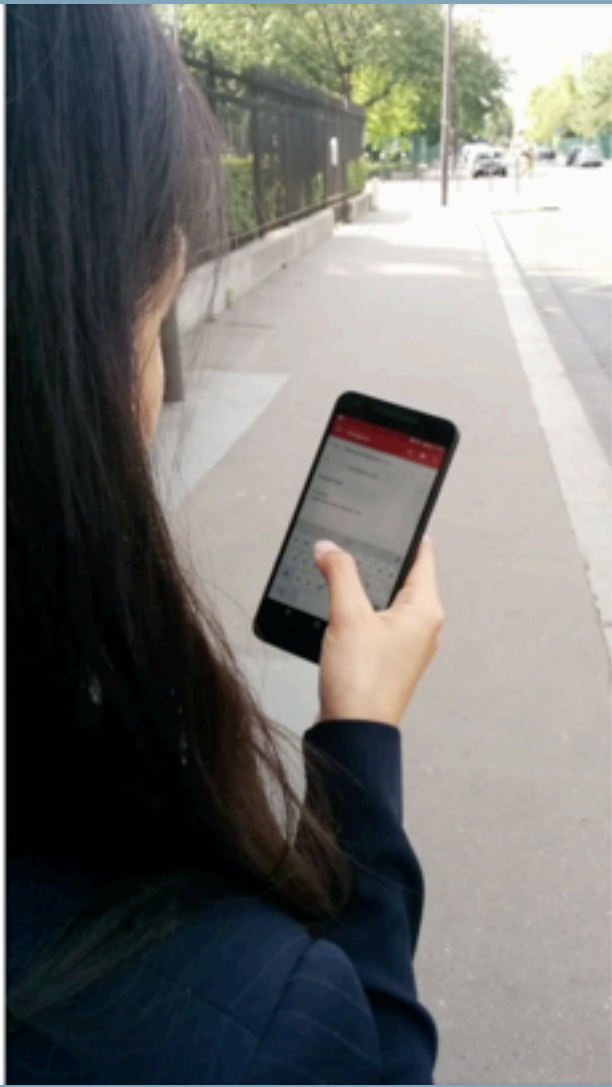
Gesture typing: Typing with gestures

Instead of tapping...

draw through
each letter
to type a word

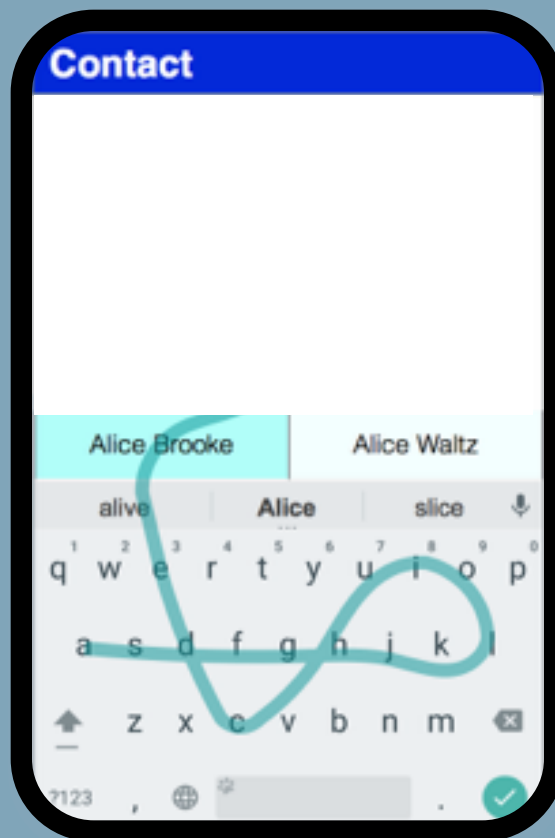


Gesture typing



CommandBoard

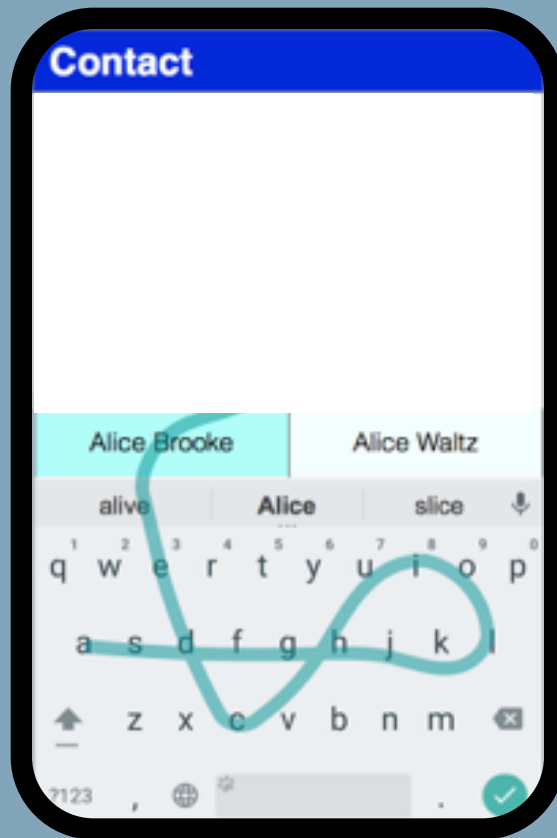
Transform gestures into commands ...



draw:
alice

CommandBoard

Transform gestures into commands ...



draw:

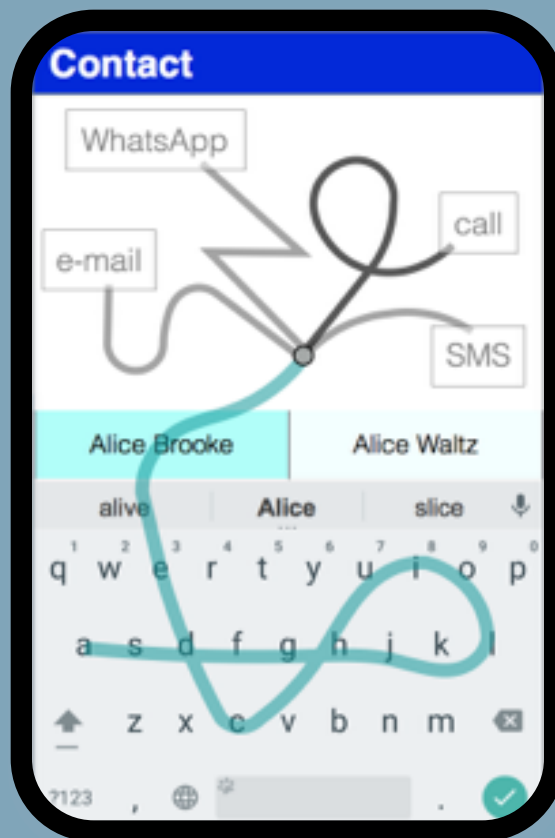
alice

choose contact:

Alice Brooke

CommandBoard

Transform gestures into commands ...



write:

alice

choose contact:

Alice Brooke

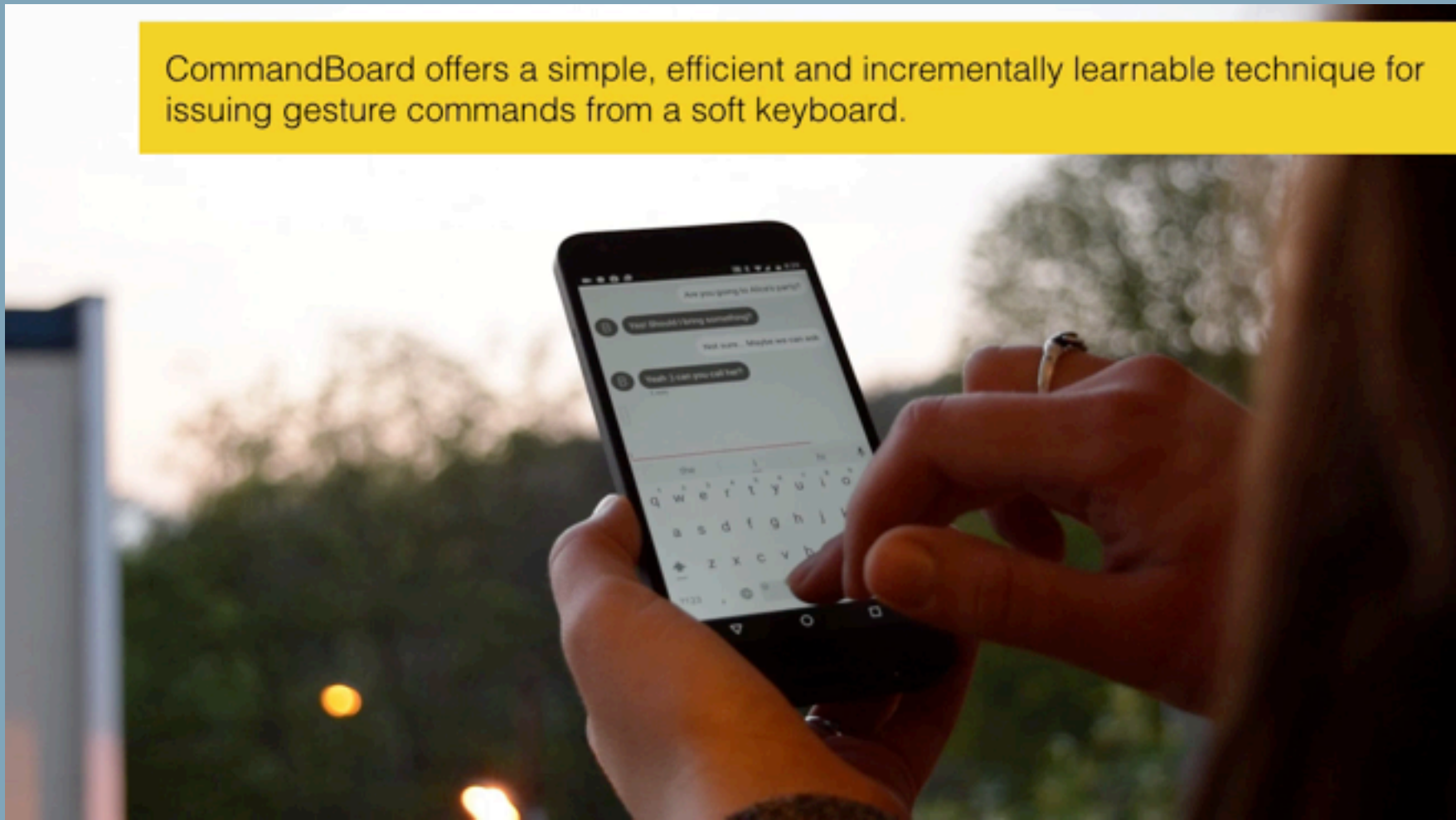
choose comm app:

Alice Brooke

CommandBoard




Command-line interaction from a soft keyboard

CommandBoard offers a simple, efficient and incrementally learnable technique for issuing gesture commands from a soft keyboard.



CommandBoard

Execute any command from a gesture keyboard

Screen Space	Contact	User Action	User Goal
<i>command-gesture input space</i>		draw gesture	execute command, add gesture shortcut
<i>command bar</i>		cross command option	choose command option
<i>suggestion bar</i>		tap word	choose word or command
<i>text input space</i>		tap key, cross key, dwell on key	enter text/emoticon, change layout, specify command

Alvina, Griggio, Bi & Mackay UIST'17

CommandBoard

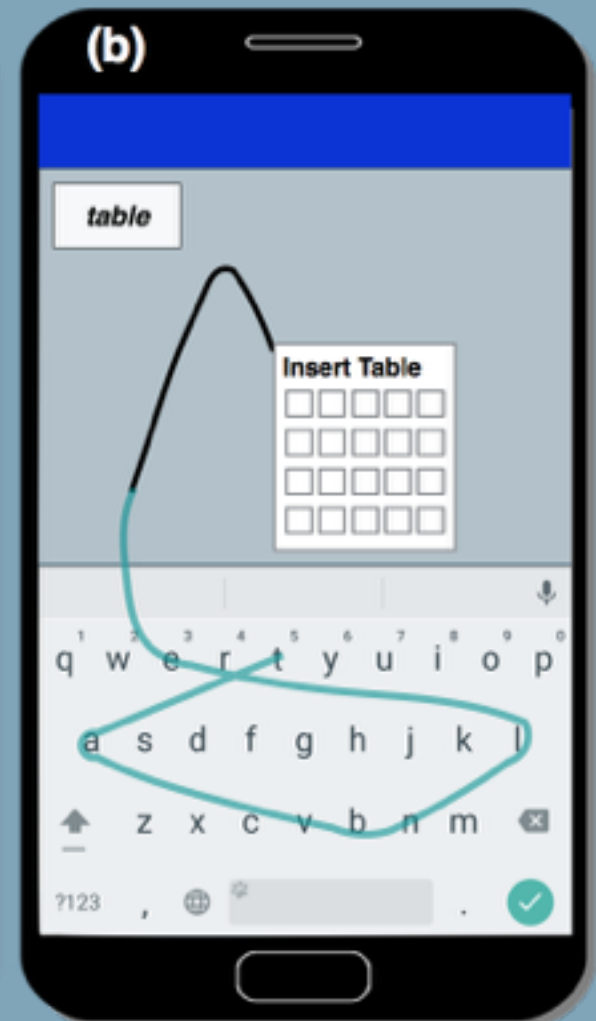
Execute any command from a gesture keyboard



CommandBoard

Draw *color*,
then pick from
a color wheel

Draw *table*,
then insert
a table





Appropriability

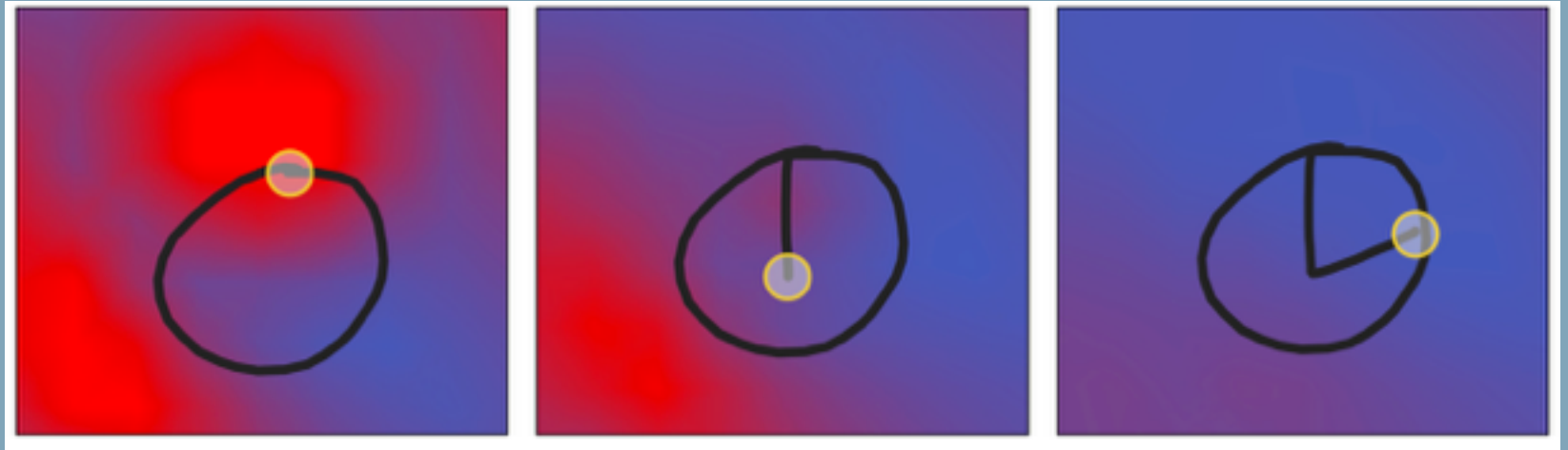
How can I
create my own
gesture commands?

Fieldward

Create your own gesture commands

Must be:

easy for you to remember



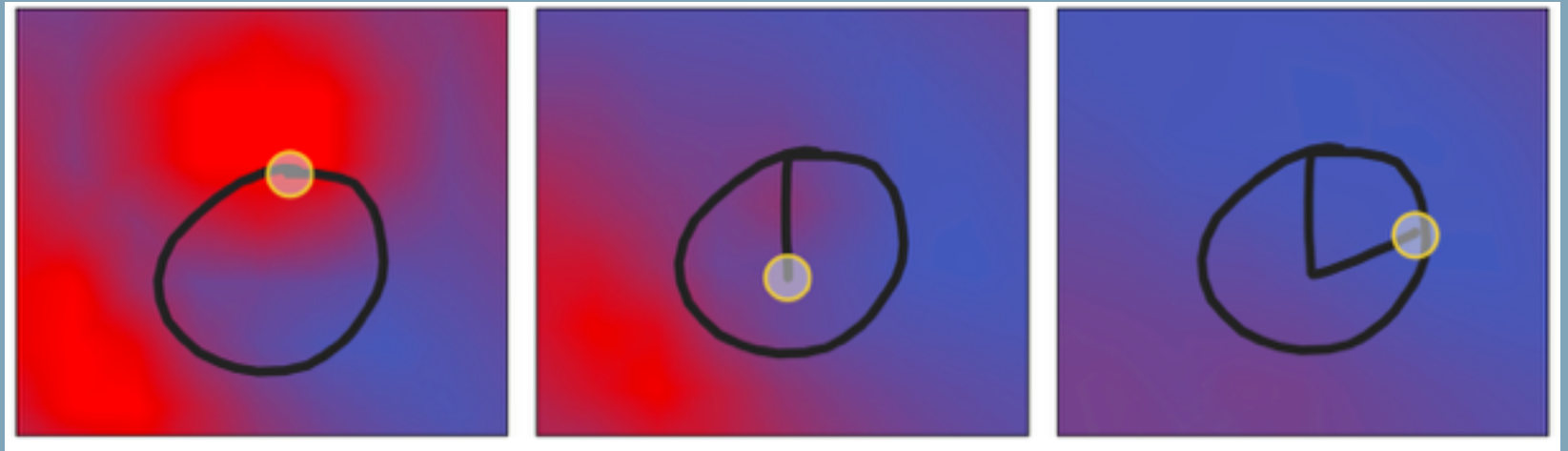
Fieldward

Create your own gesture commands

Must be:

- easy for you to remember

- easy for the system to recognize

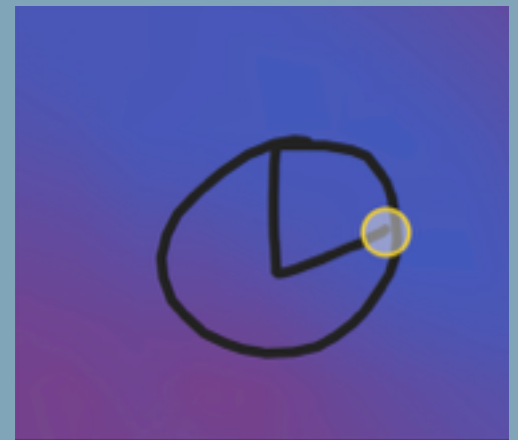
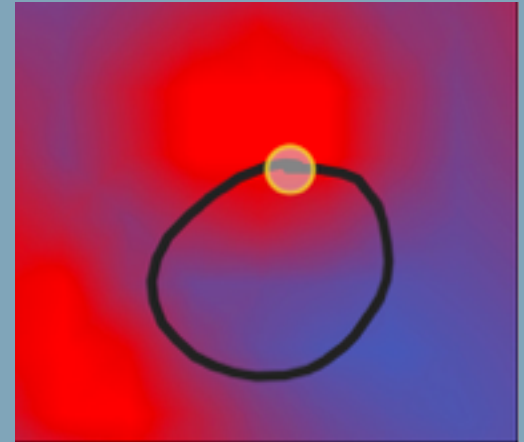


Fieldward

Draw a gesture

If it ends in a red zone
the gesture is ambiguous

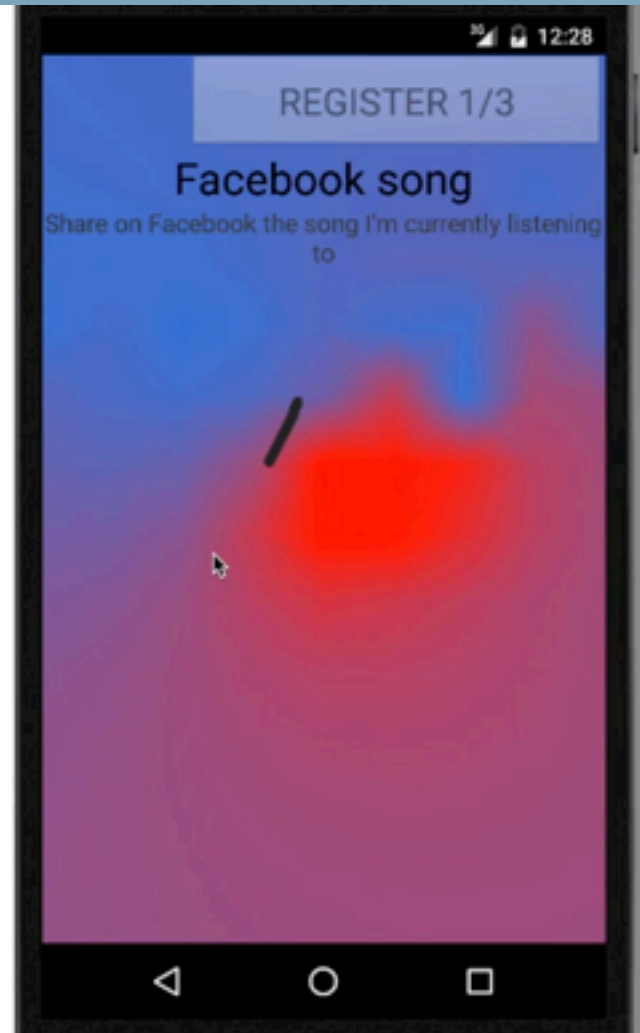
If it ends in a blue zone
you have a new gesture !



Fieldward:

Fieldward

Shows a color gradient indicating optimal directions to make a recognizable gesture



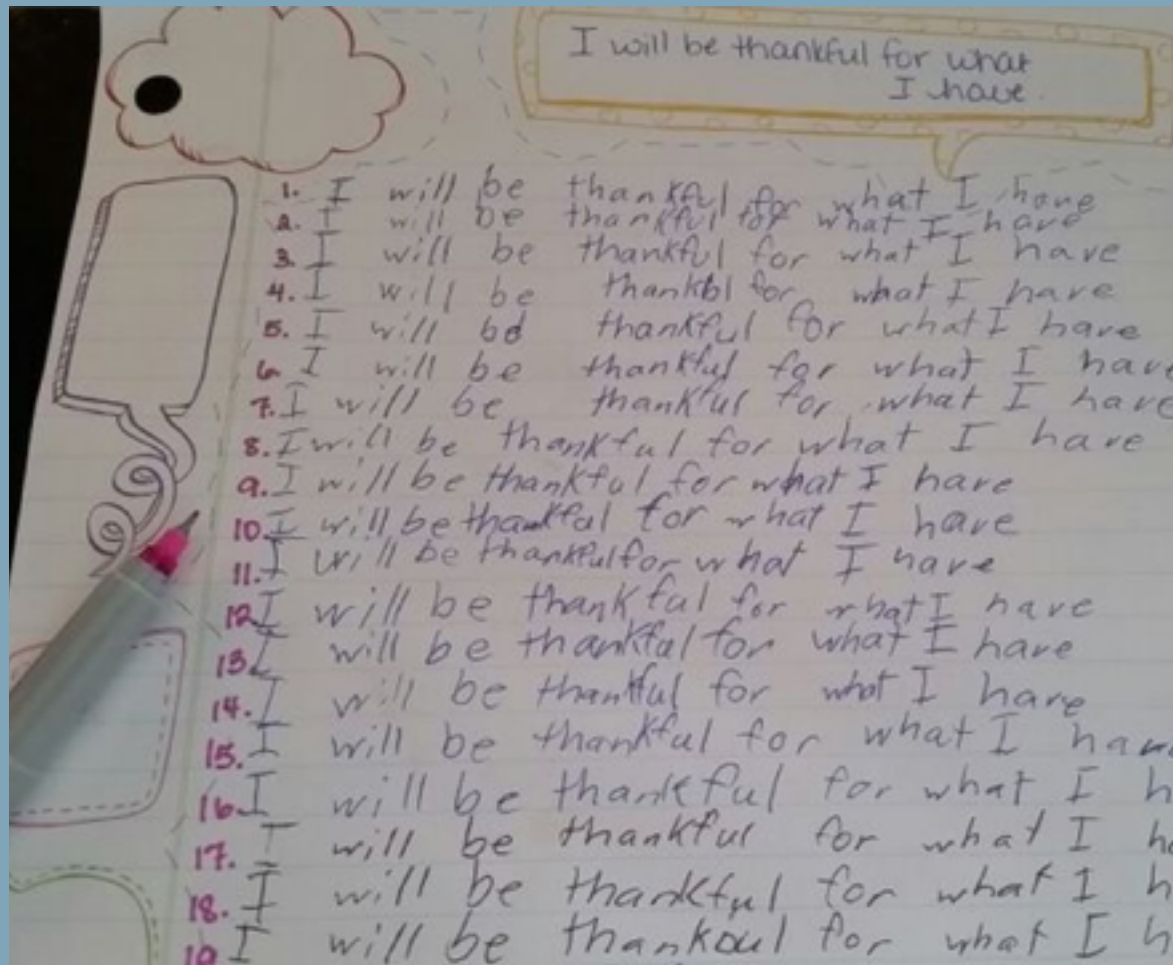


Expressivity

How can I
generate
expressive text?

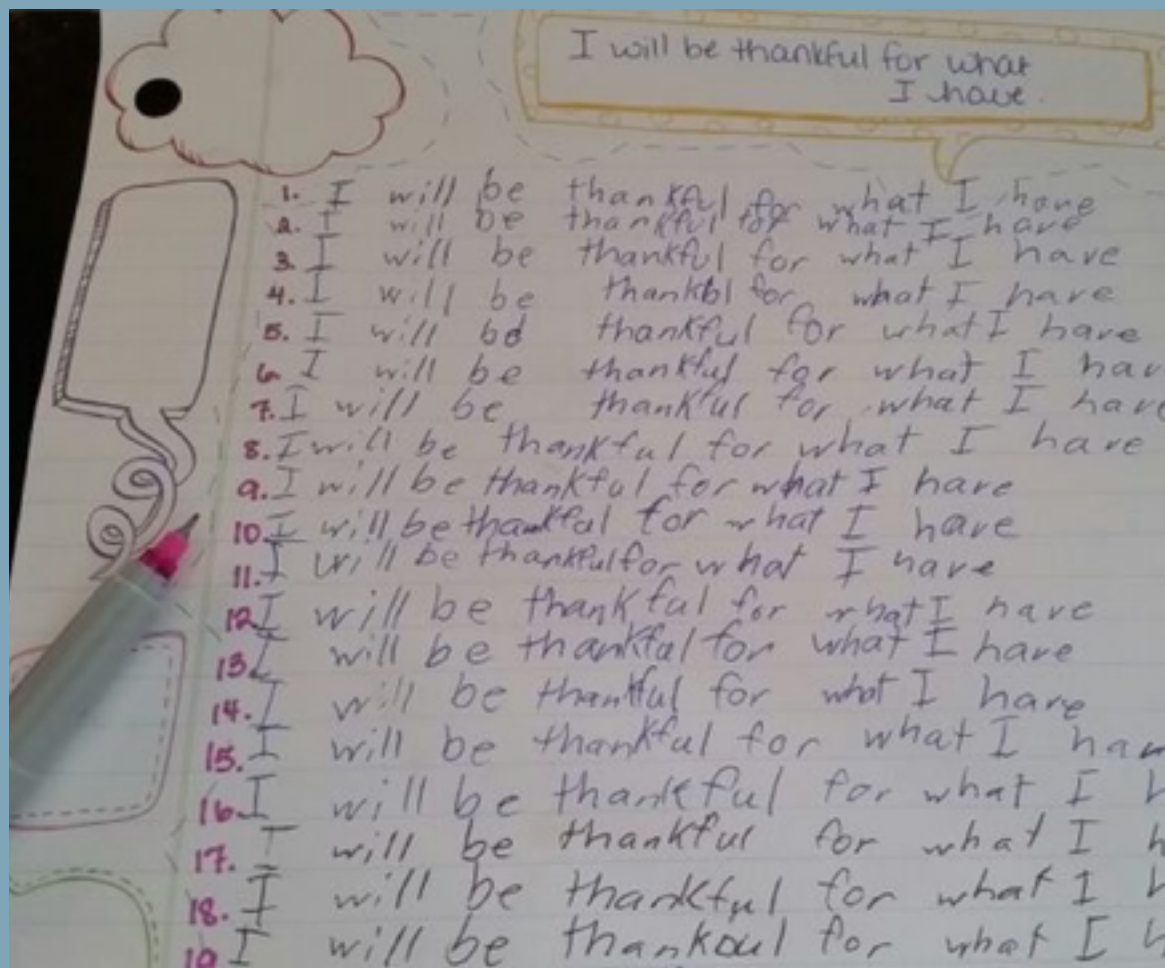
Human expression

Handwriting is expressive



Human expression

Handwriting is expressive ... SMS messages -- not so much



Human expression vs. Machine classification

Machine learning algorithms:

- Goal is to classify the correct word

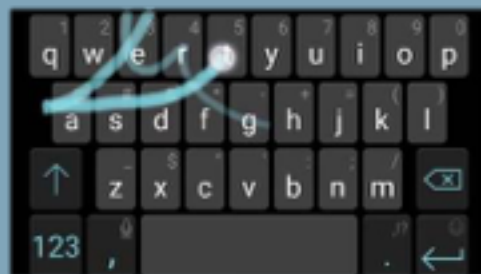
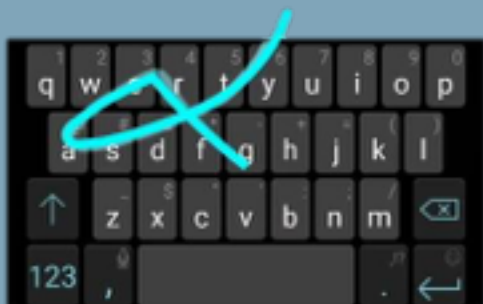
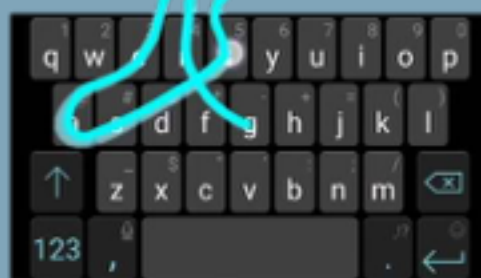
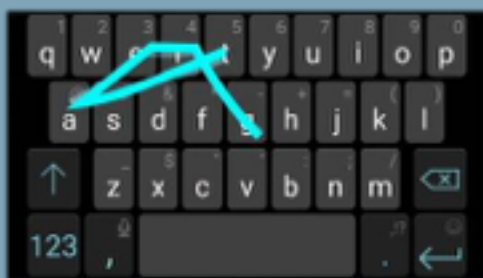
- Human variation is treated as noise

Gesture typing algorithms are great . . .



Human expression vs. Machine classification

Four ways to input the word “great”



All produce the identical result: **great**

Expressive Keyboard vs. Machine classification

Machine learning

- Guess the correct word (classify)

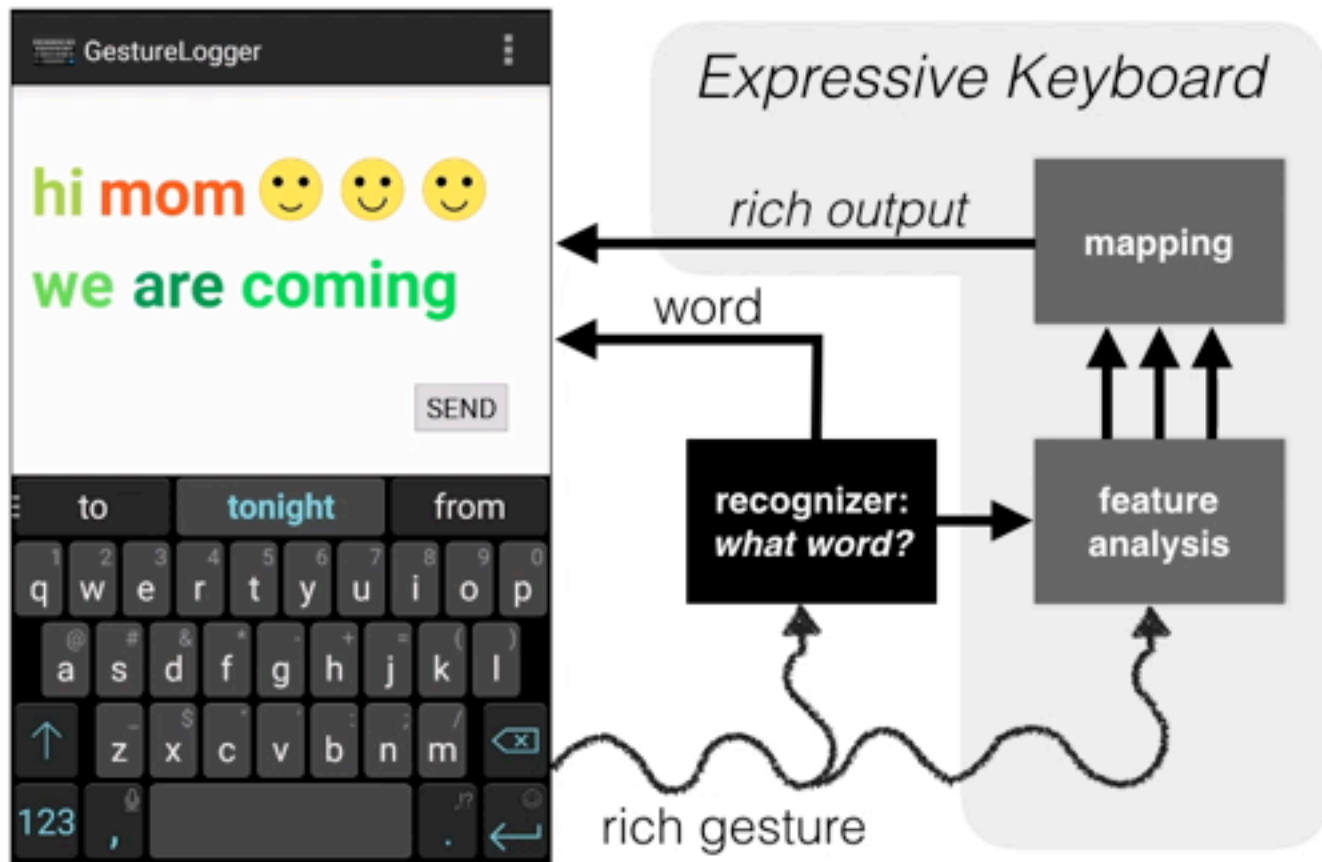
- Throw away human variation

Human-centered approach

- Create expressive output

- Transform human variation

Expressive Keyboard

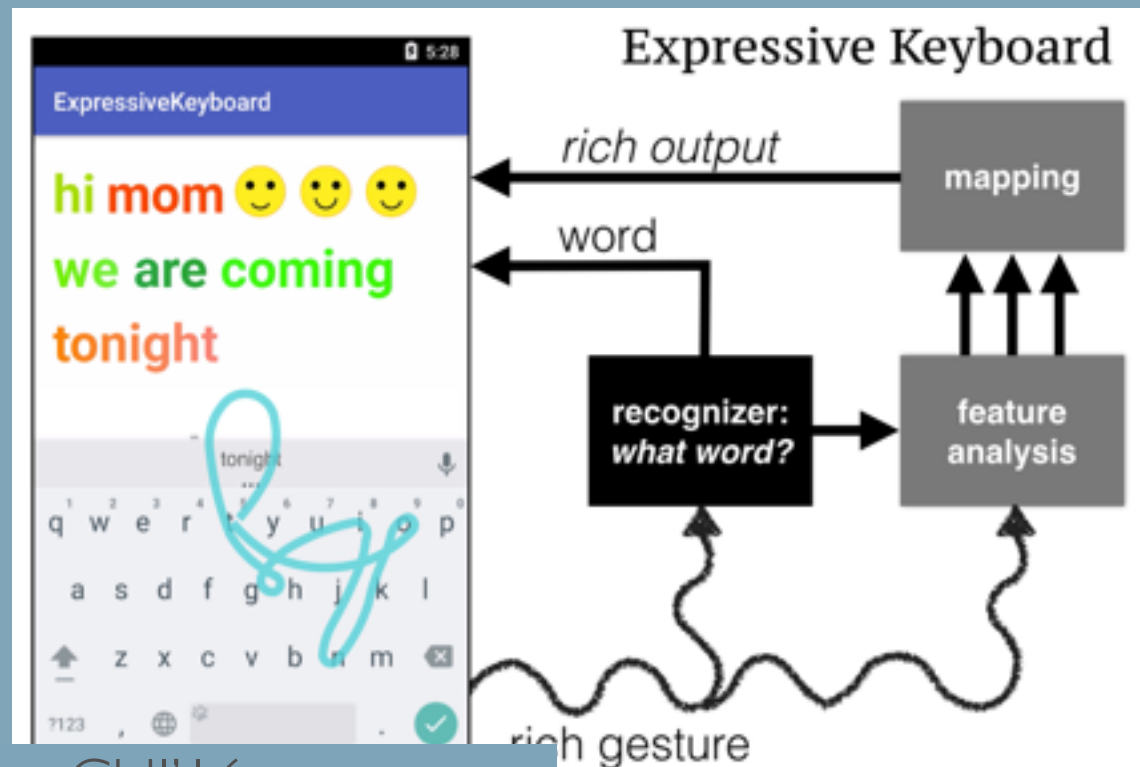


These features can be mapped to color,
change the expression of an emoji,

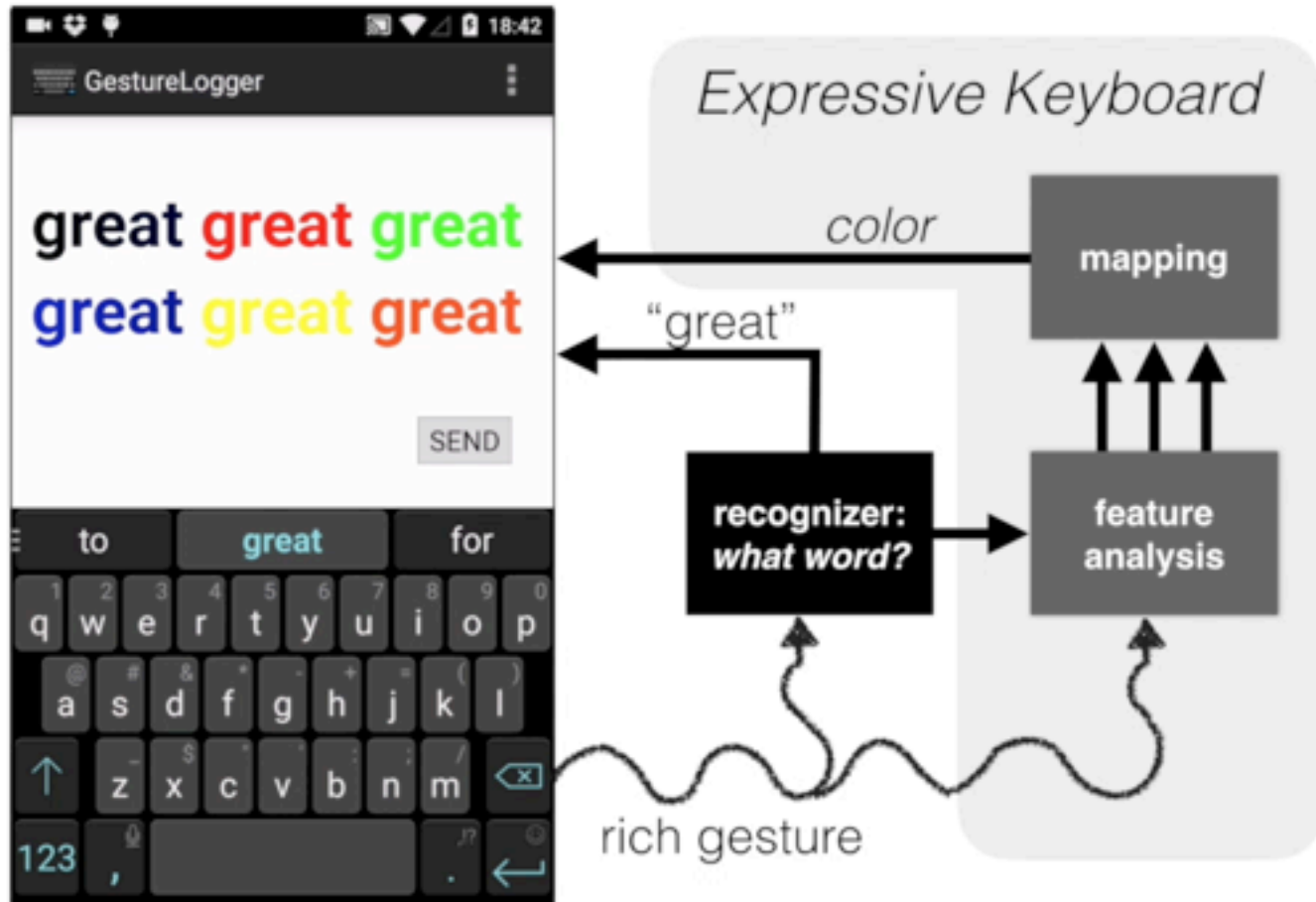
Expressive Keyboard

Map gesture variation to output properties

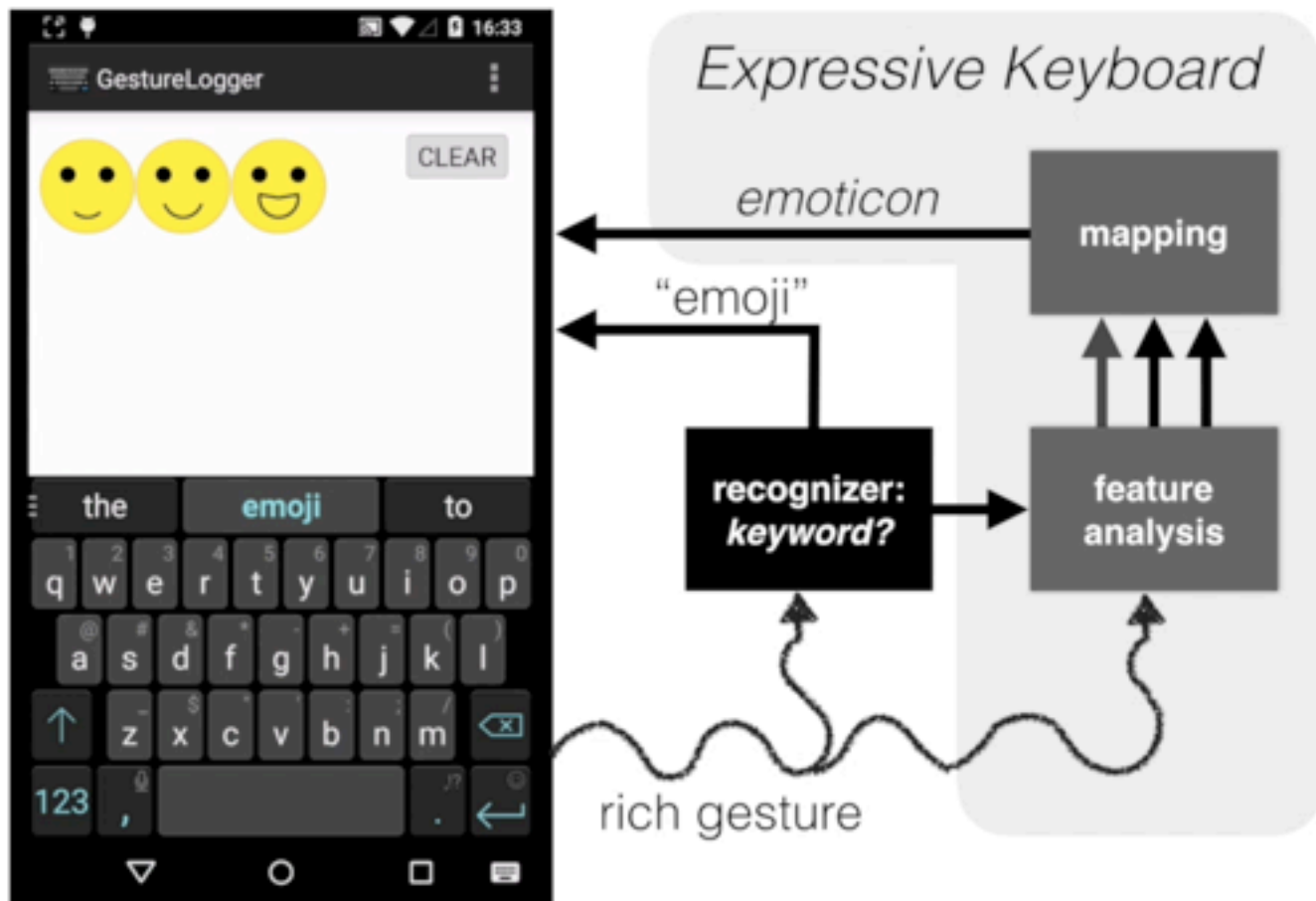
Users control:
text color
font style
emojis



Expressive Keyboard – measure variation



Expressive Keyboard – Expressive emojis



Next: Expressive Typography

Current project:

Express how the text appears ... by modifying your gesture typing

Expressive Typography project. Contexts for using expressive typography. Ways to control input and output features in different contexts. Demonstration of the prototype and the font variations. Types of dynamic font variations. The quick brown fox jumps over the lazy dog. Sphinx of black quartz, judge my vow. (Plain)

Expressive Typography project. Contexts for using expressive typography. Ways to control input and output features in different contexts. Demonstration of the prototype and the font variations. Types of dynamic font variations. The quick brown fox jumps over the lazy dog. Sphinx of black quartz, judge my vow. (Informal-Totally)

Expressive Typography project. Contexts for using expressive typography. Ways to control input and output features in different contexts. Demonstration of the prototype and the font variations. Types of dynamic font variations. The quick brown fox jumps over the lazy dog. Sphinx of black quartz, judge my vow. (Spread-Totally)

Expressive Typography project. Contexts for using expressive typography. Ways to control input and output features in different contexts. Demonstration of the prototype and the font variations. Types of dynamic font variations. The quick brown fox jumps over the lazy dog. Sphinx of black quartz, judge my vow. (Elegant-Totally).

Tomorrow:
true human-
computer
partnerships
that empower
rather than
frustrate
(or replace)
people

