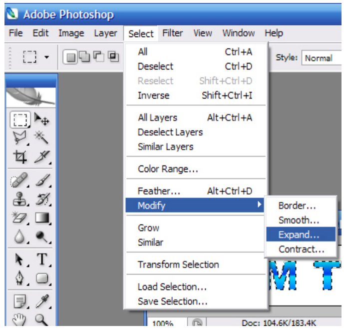
Week 4 – Part 1

Shortcuts, gestures, phrasing & chunking interaction, crossing interfaces

Menus: strengths

Recognition vs. recall

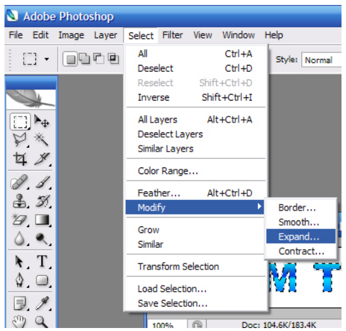
Exploratory, incremental learning



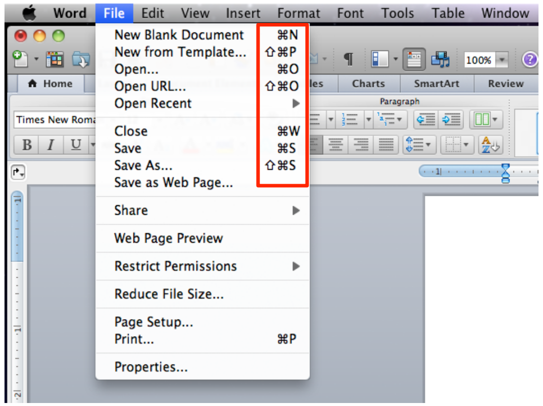
Menus: drawbacks

Slow and tedious

Inappropriate for repetitive actions



Keyboard shortcuts (hotkeys)



Keyboard shortcuts (hotkeys)

« 251 experienced users of Microsoft Word were given a questionnaire assessing their choice of methods for the most frequently occurring commands. Contrary to our expectations, most experienced users rarely used the efficient keyboard shortcuts, favoring the use of icon toolbars instead.»

Lane et al. (2006)

« While our participants stated a strong preference for keyboard shortcuts and reported far more shortcut usage than did the less experienced users studied by Lane et al., shortcuts still had a fairly low usage. »

Hendy et al. (2010)

Improving hotkey learning
(Grossman et al. 2007)

Two main problems of hotkeys
- **Hard to learn** - selecting menus and using hotkeys are radically different actions (not clear mapping)
- **Lack visibility**

Possible improvements
- Increase their **exposure**
- Call for user **attention**
- Support **incidental learning**
- Enrich **presentation modalities**

Improving hotkey learning
(Grossman et al. 2007)

Proposed designs

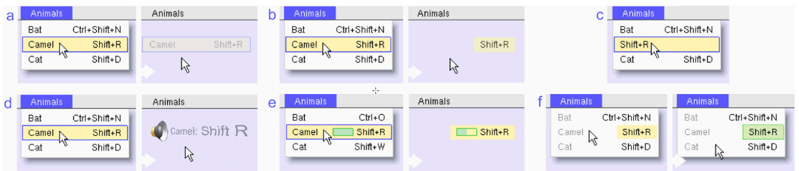


Figure 2. Examples of designs emphasizing hotkey mappings. (a) Traditional design. (b) Fading-out hotkey. (c) Hotkey menu replacement. (d) Audio feedback. (e) System delay. (f) Disabled menu items. Refer to text for detailed description.

An experiment showed that using audio feedback and disabling menu items can accelerate learning

Improving hotkey learning
(Grossman et al. 2007)

Proposed designs

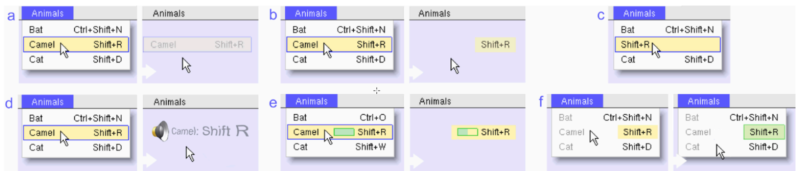
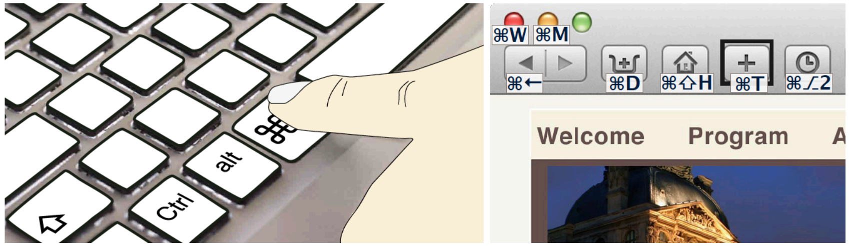


Figure 2. Examples of designs emphasizing hotkey mappings. (a) Traditional design. (b) Fading-out hotkey. (c) Hotkey menu replacement. (d) Audio feedback. (e) System delay. (f) Disabled menu items. Refer to text for detailed description.

Limitations of these solutions?

ExposeHK (Malacria et al. 2013)

Feedforward as soon as the user presses a modifier key



http://www.gillesbailly.fr/hotkeys.html

Gestures

« A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed.»

(Kurtenbach & Hulteen, 1990)

Others consider simple button presses as **zero-degree** gestures (Zhai et al., 2012)

Taxonomies of gestures

Semiotic: used to communicate meaningful information

Ergotic: used to manipulate the physical world and create artifacts

Epistemic: used to learn from the environment through tactile or haptic feedback

Cadoz (1994)

Semiotic gestures

Symbolic: they bear a single meaning (e.g., wave the hand to say « hello » or « bye »)

Deictic: pointing gestures, directing attention to specific objects or events (e.g., place it « *there* »)

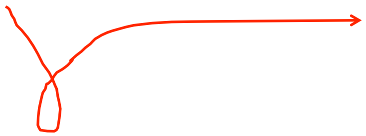
Iconic: convey information about the size, shape or orientation of an object (e.g., it moved like « this »)

Rime & Schiaratura (1994)

2D strokes as gestures

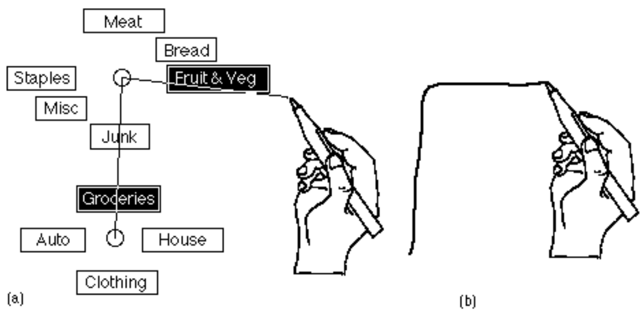
Defined as a sequence of data points (x_i, y_i, t_i)

where timestamps can be used to capture the dynamics of a gesture (e.g., local or global velocity and acceleration)



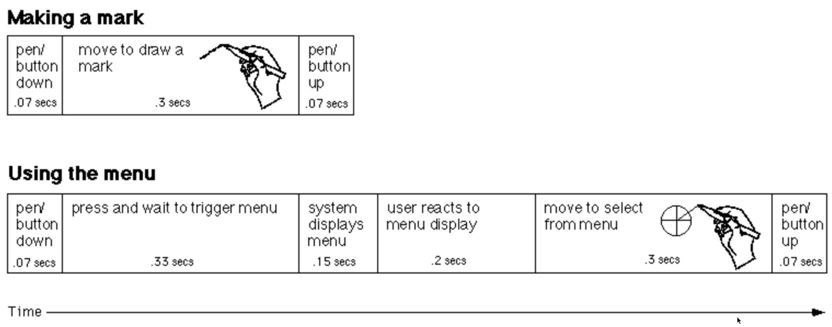
Input devices

Marking menus (Kurtenbach & Buxton, 93)

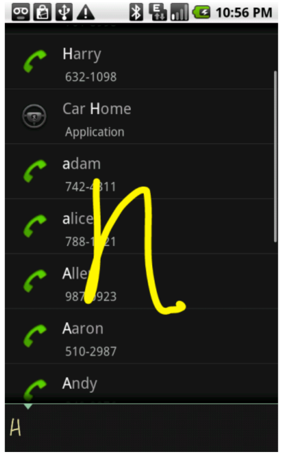


Novice use (pause for feedforward) Expert use

Marking menus (Kurtenbach & Buxton, 93)



Symbolic strokes

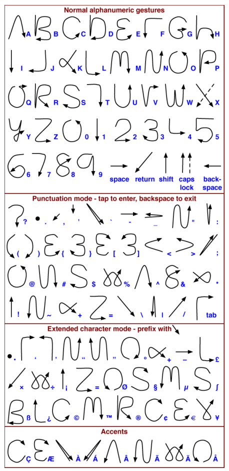


Gestural strokes for mobile search (Li, 2009)

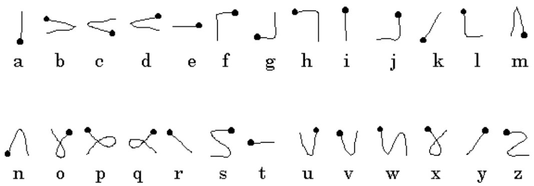
Gestures vs. hotkeys

Better to learn and recall gestural shortcuts (Appert & Zhai, 2009)

Stroke alphabets



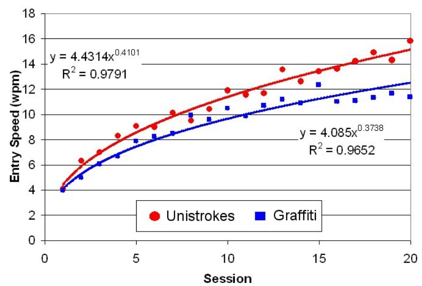
Graffiti (Palm OS)

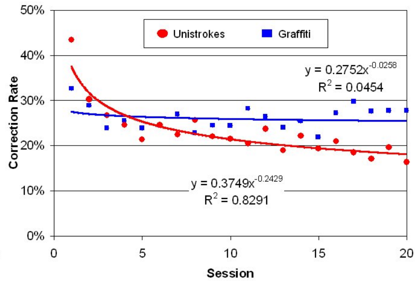


Unistrokes (Xerox PARC)

Graffiti vs. Unistrokes

Graffiti is easier to learn (closer to Latin) but slower than Unistrokes

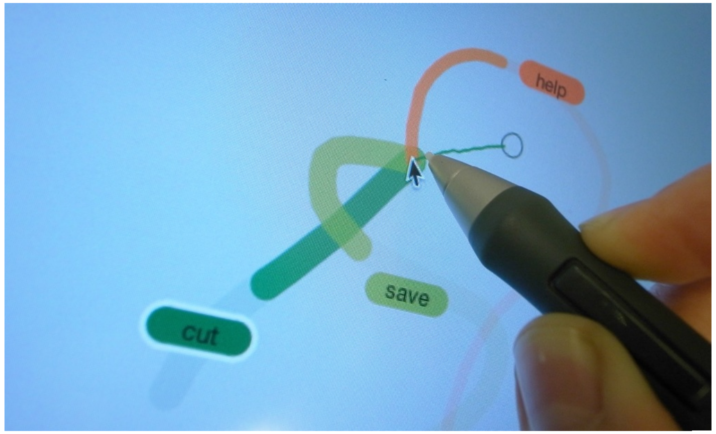




10 participants, 20 sessions during 6 weeks (Gastellucci & McKenzie, 2008)

Support feed-forward & learning

OctoPocus (Bau & Mackay, 2008)

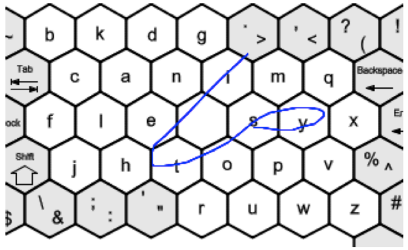


http://www.olivierbau.com/octopocus.php

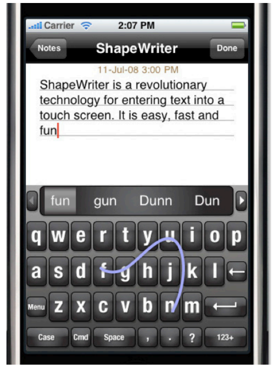
Gestural text entry

SHARK/ShapeWriter (Zhai et al., 2003-)

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

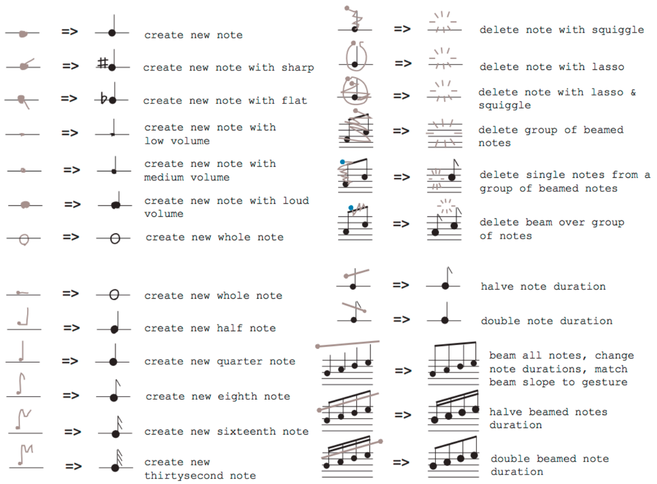


ATOMIK layout (optimized for performance)



ShapeWriter on iPhone (2008)

Music Notepad (Forsberg et al, 1998)



https://www.youtube.com/watch?v=p-jMKqAPrOs

Gesture recognition

Several common learning-based classification techniques can be used, e.g. k-nearest neighbor, support vector machines

- Recognition is based on the use of a training set that provides samples of the gestures of interest

Some terminology:

True positives: gestures correctly classified under a given class

False positives: gestures falsely classified under a given class

True negatives: gestures correctly not included under a given class

False negatives: gestures incorrectly not included under a given class

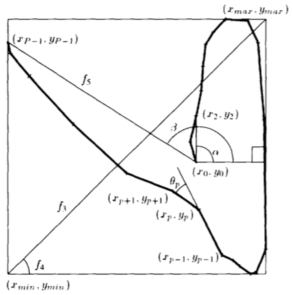
Rubine’s recognizer (Rubine, 1991)

Simple, requiring a relatively small number of samples

Provides measure for avoiding false positives

Recognition is based on a range of distinctive stroke features

- initial angle
- angle and length of bounding box
- distance between first and last point
- total angle
- maximum speed
- duration of the gesture
- etc.



Rubine’s recognizer (Rubine, 1991)

Implementations in Java

iGestures: http://www.igesture.org/algo_rubine.html

JavaSwing: http://swingstates.sourceforge.net

Tutorial page by Géry Casiez (text in French):
http://www.lifl.fr/~casiez/IHM/TP/TP6Rubine/

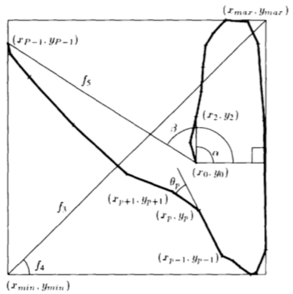
Rubine’s recognizer

Simple, requiring a relatively small number of samples

Provides measure for avoiding false positives

Recognition is based on a range of distinctive stroke features

- initial angle
- angle and length of bounding box
- distance between first and last point
- total angle
- maximum speed
- duration of the gesture
- etc.



Dollar recognizers (Wobbrock et al.)

Easy-to-deploy recognizer, designed for rapid prototyping

$1 (2007): one-stroke gestures, about 100 lines of code
https://depts.washington.edu/aimgroup/proj/dollar/

Protractor (2010): improves speed and accuracy of $1 recognizer

$N (2010): multistroke recognizer
https://depts.washington.edu/aimgroup/proj/dollar/ndollar.html

$P (2012): most recent recognizer for both unistrokes and multistrokes, better performance
https://depts.washington.edu/aimgroup/proj/dollar/pdollar.html

Chunking & Phrasing (Buxton, 1986)

Tasks are often compound

Example: menu selection
- (1) Invoke the menu
- (2) Navigate to selection
- (3) Make selection and return

There is an underlying grammar that determines how subtasks are « glued » together

http://www.dgp.toronto.edu/OTP/papers/bill.buxton/chunking.html

Chunking & Phrasing (Buxton, 1986)

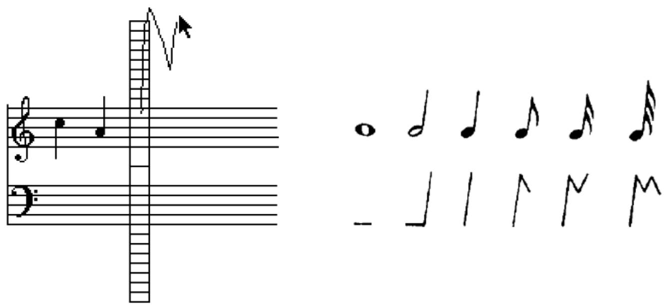
Challenges
- How to make an interaction grammar visible to the user?
- How to « glue » the partial actions of a task together to avoid errors?

Chunking & Phrasing (Buxton, 1986)

Example
- The glue of activating and selecting an item through a pie menu is the tension of the finger, which stays pressed throughout the whole selection process.

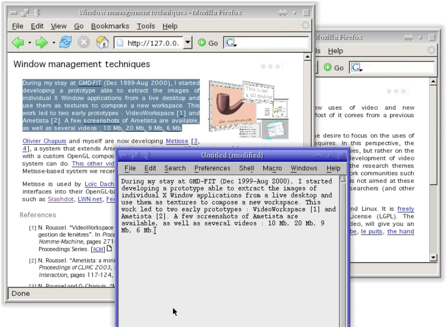
Chunking & Phrasing (Buxton, 1986)

Other example: specifying the position and type of the note with a unique gesture



Phrasing interaction

Phrasing an interaction sequence sometimes requires imagination and innovation!

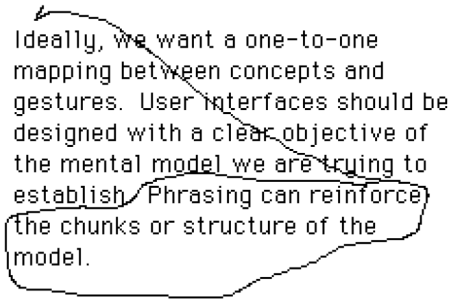


Copy-paste between overlapping windows (Chapuis & Roussel, 2007)
http://insitu.lri.fr/metisse/rock-n-roll/

Scope of a gesture

Moving text from (Buxton, 1986)

Scope of the action's target



scope of the action's source

Gesture components

Identifier of the command/action

Source & target scope of the gesture (optional)

Additional parameters (optional)

- e.g., target size of the object of interest

More complex example

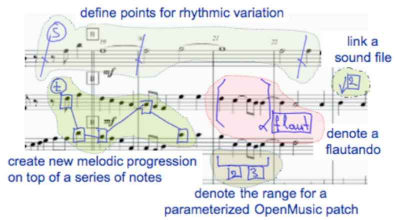


Figure 5. Examples of operations expressed with Mus*ink*

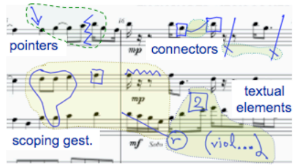


Figure 6. Examples of Mus*ink* basic gestures

Mus*ink* (2009)

Delimiters

CANYOUTELLMEWHATYOUREADHERE?

Delimiters

CANYOUTELLMEWHATYOUREADHERE?

CAN YOU TELL ME WHAT YOU READ HERE?

Delimiters

Systems does not think like users. They need help to be able to chunk gestures into their partial components:

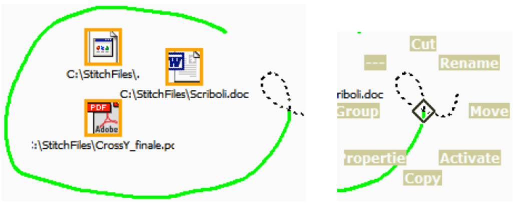
- e.g., which part is the scope and which part is the gesture identifier?

Recognizers have a limited scope and make mistakes. You should use them sparingly.

As in language grammars, we need "punctuation marks" that chunk interaction sequences into partial actions

Delimiters (Hinkley et al, 2005)

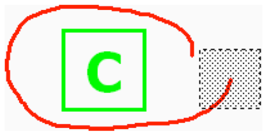
Separating the scope from the command part of a gesture:
- (a) select a group of objects with a lasso
- (b) use a marking menu to apply a command, e.g, delete



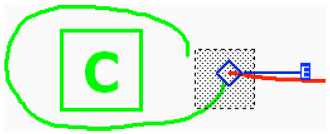
Using a **pigtail** as delimiter

Delimiters (Hinkley et al, 2005)

Separating the scope from the command part of a gesture:
(a) select a group of objects with a lasso
(b) use a marking menu to apply a command, e.g, delete



Show a small rectangular **handle** when lifting the pen



Put the pen inside the **handle** to start the marking gesture

Delimiters (Hinkley et al, 2005)

They compared four delimiting techniques:
(1) pigtail
(2) handle
(3) time-out (time threshold after pausing the pen)
(4) button press

Results

- Button press generated the most errors
- Handle resulted in less errors and was the most preferred technique
- Pigtail was slightly faster in repeated trials

Switching modes

How to differentiate between regular drawing or writing and command gestures?

Li et al. (2005) compared five approaches
- Pressing button on stylus
- Pressing button with non-dominant hand
- Pressing and holding to wait for mode change
- Press with different force level to switch mode
- Use the eraser tip of the pen for gestures

Switching modes

How to differentiate between regular drawing or writing and command gestures?

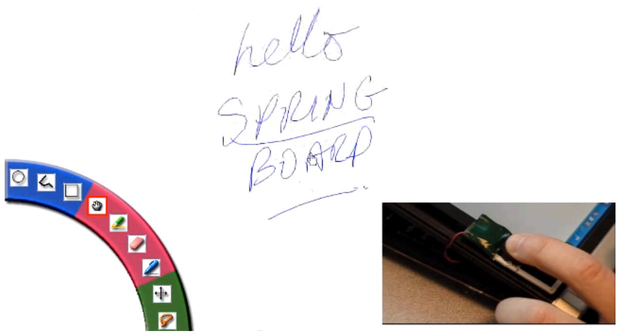
Li et al. (2005) compared five approaches
- Pressing button on stylus
- Pressing button with non-dominant hand
- Pressing and holding to wait for mode change
- Press with different force level to switch mode
- Use the eraser tip of the pen for gestures

They found that pressing a button with the non-dominant hand was the fastest and most preferred approach

Springboard (Hinckley et al 2006)

Enrich the set of available quasimodes by combining the use of a button (non-dominant hand) and a pallette of tools

The tool remains activated as long as the user keeps the button pressed



Crossing instead of pointing

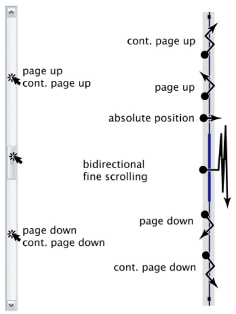
(Accot & Zhai, 2002)

(a) Pointing a target

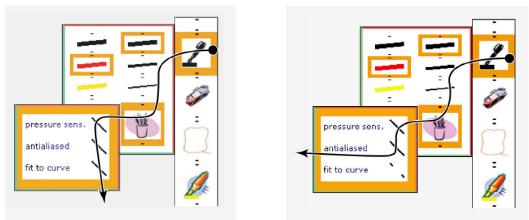
(b) Crossing a goal

Crossing interfaces

CrossY (Apitz & Guimbretière, 2004)



Crossing-based gestures for navigation

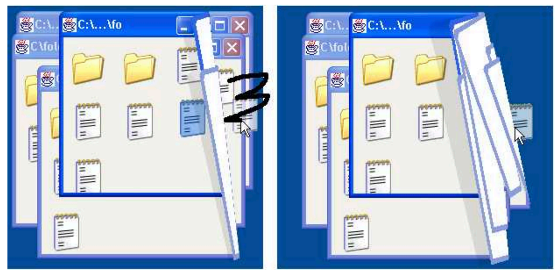


Crossing-based selection

http://www.cs.umd.edu/hcil/crossy/

Crossing interfaces

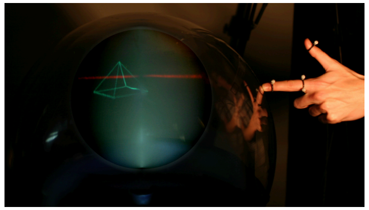
Fold n' Drop (Dragicevic, 2004)



https://www.lri.fr/~dragice/foldndrop/

Gestures in following classes

Multitouch gestures



Free-hand gestures