

# Instrumented Environments

Andreas Butz, [butz@ifi.lmu.de](mailto:butz@ifi.lmu.de), [www.mimuc.de](http://www.mimuc.de)



# Low level context recognition

..on the other end of the scale..

# Low level context recognition

(Cakmakci et al. 2002)

- Design “context aware hardware”
- Enhance wearable computing
- Detect simple user activities, like sitting, walking, looking at the watch



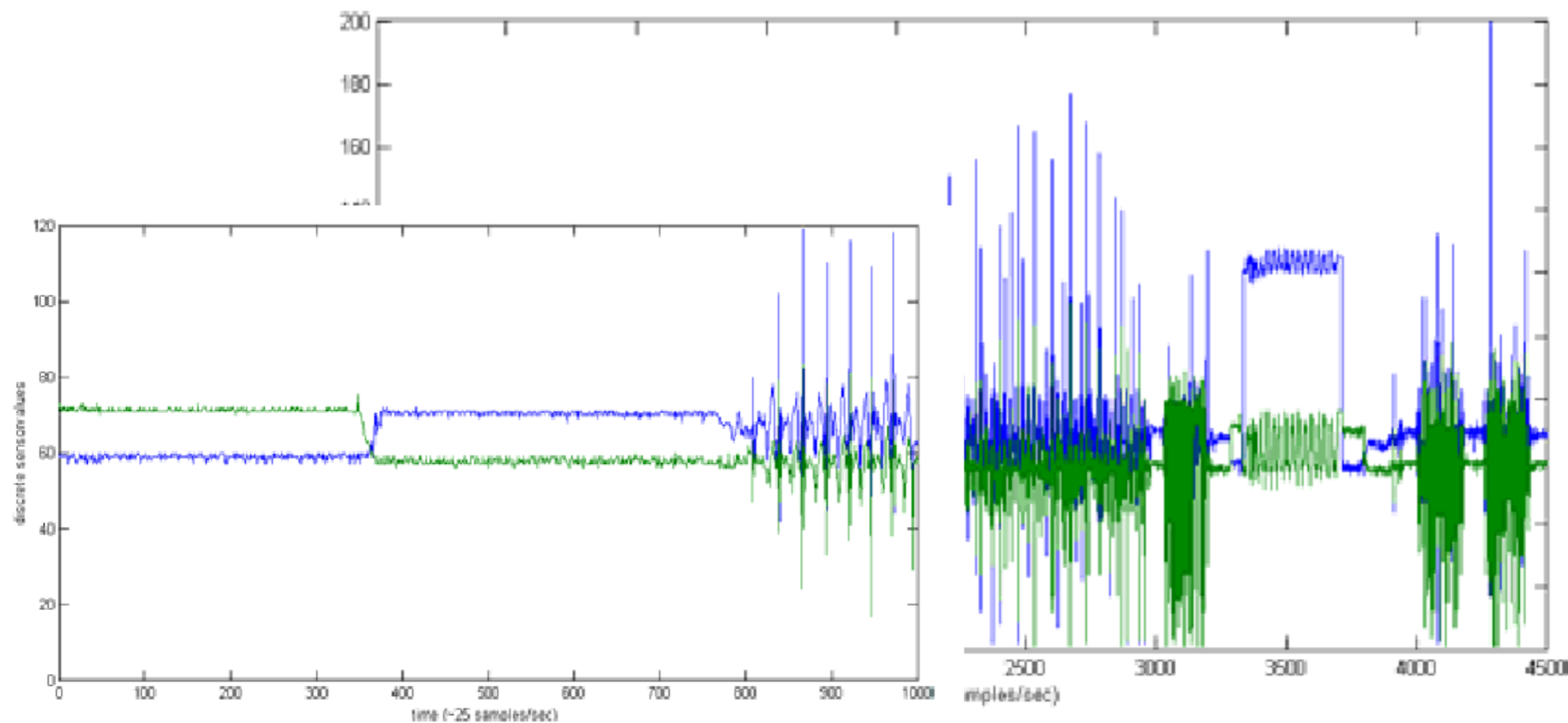
# Low level context recognition

- Use statistical modeling techniques from robotics to determine context
- Use accelerometers to record movement changes
- Apply Bayes rule to determine probability of certain contexts:

$$p(\textit{context} \mid \textit{sensordata}) = \frac{p(\textit{sensordata} \mid \textit{context}) * p(\textit{context})}{p(\textit{sensordata})}$$

# Low level context recognition

- First simple Experiment: Detect whether users are sitting, standing and walking



Acceleration over 4500 data points during the experiment  
Use the first 1000 data points for learning

# Low level context recognition

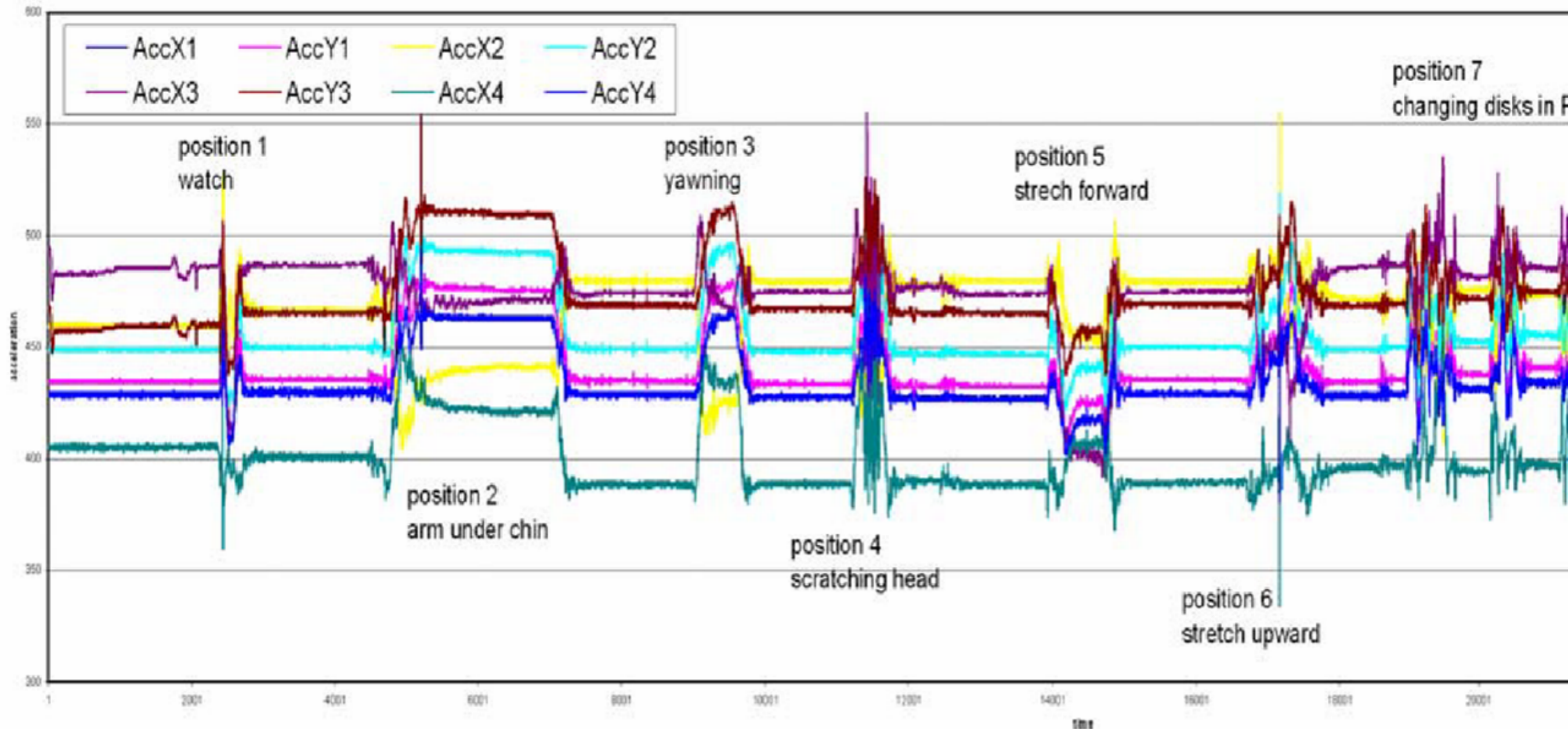
- Results of recognition

Activity	Recognition rate
Sitting (occurs 3 times during the experiment)	95.66%
Standing (occurs 2 times during the experiment)	80%
Walking (occurs 9 times during the experiment)	93.11%

# Low level context recognition

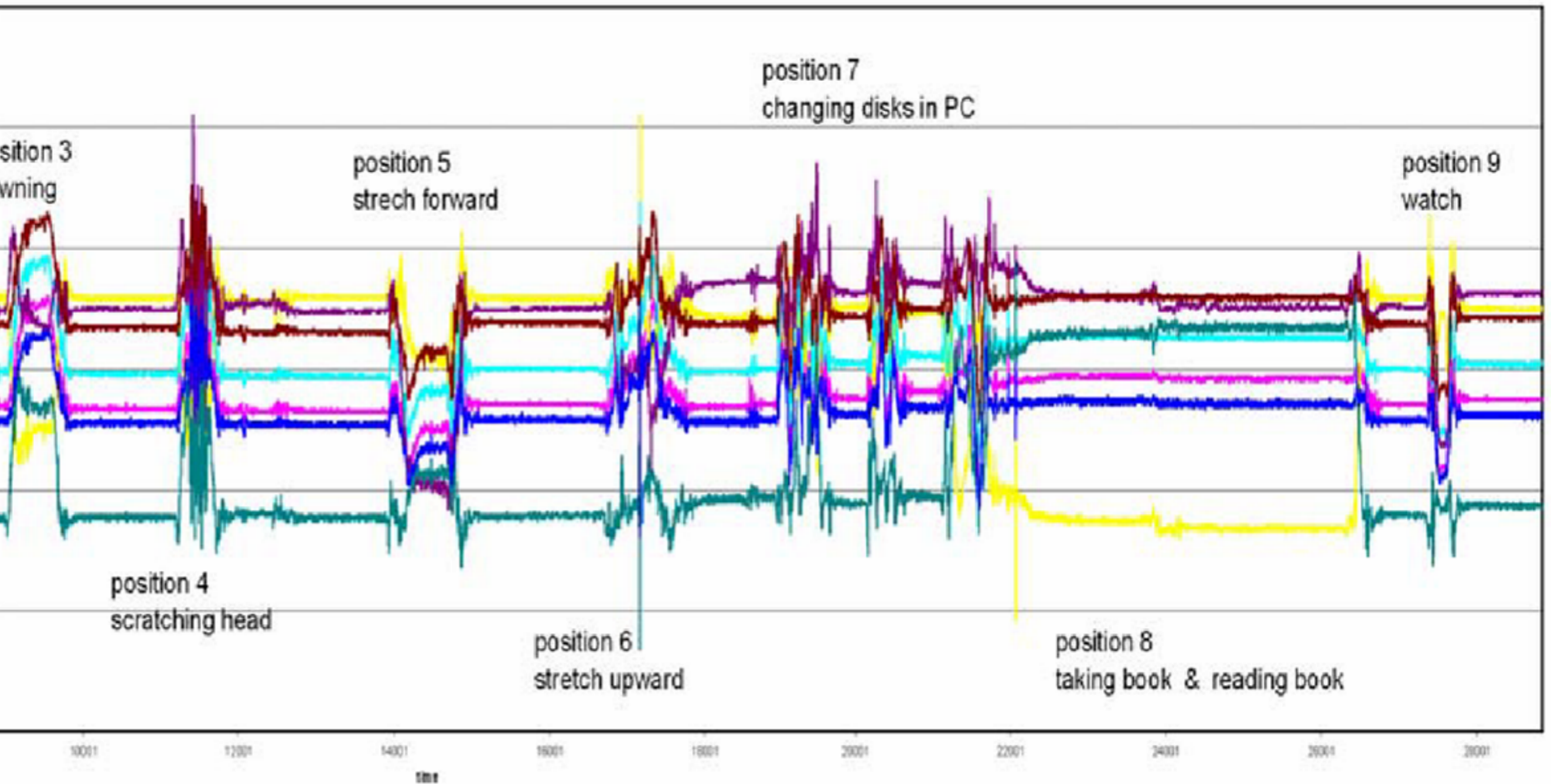
- 2nd experiment: Detect when users glance at their watch
- Idea: reduce power of watch (e.g. toggle display) during use
- Extremely important for a wristwatch computer (IBM WWC: ARM7 processor, 8Mb flash memory, 8Mb of DRAM, serial, IRDA, and expansion interfaces).

# Low level context recognition

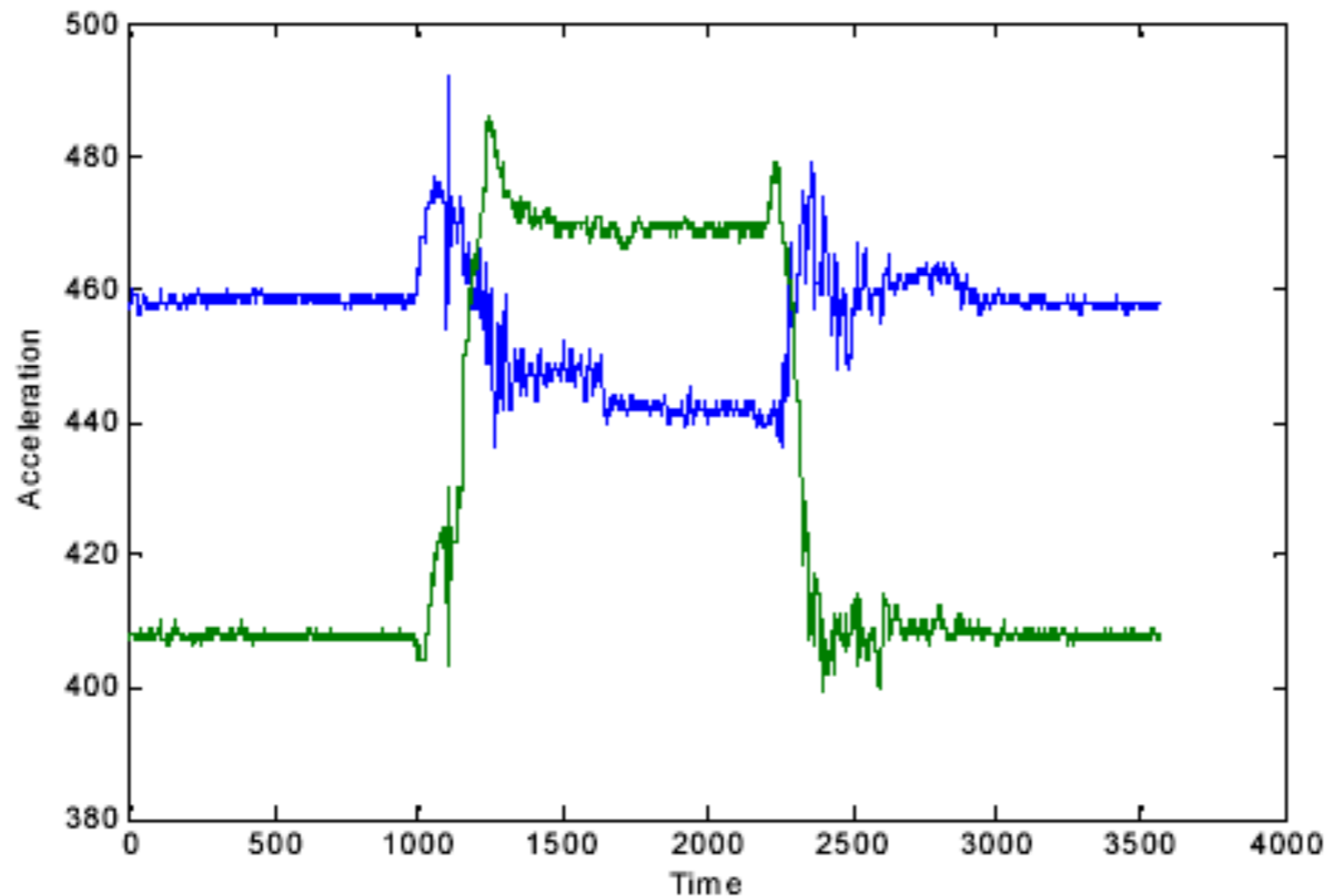


Evaluation set in a blind experiment





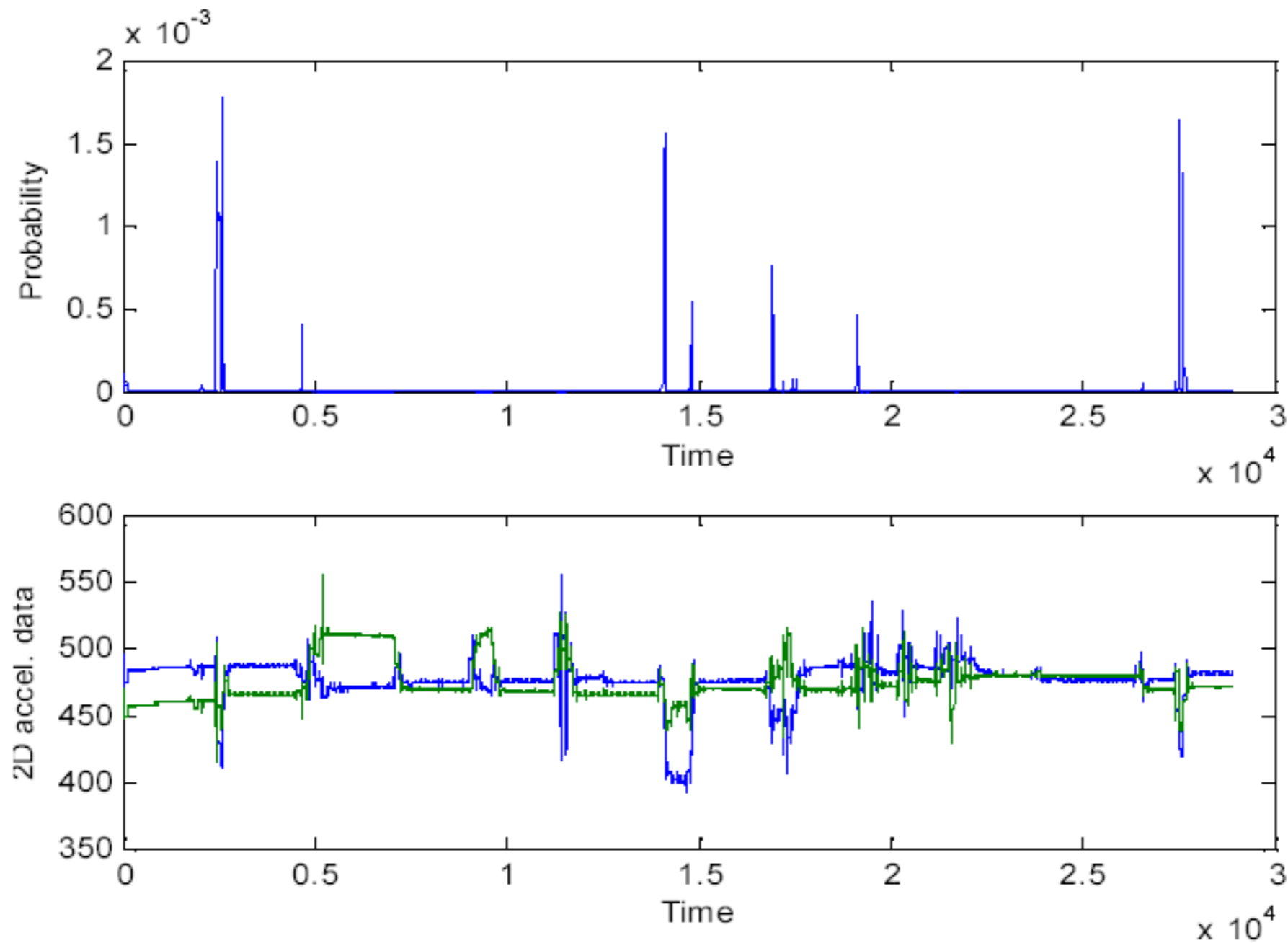
# Low level context recognition



**Figure 4.** A typical example of training data for looking the watch gesture.

Use a single hypothesis approach to model “wrist watching”  
(Only one well defined class)

# Low level context recognition



Results:

Probability of certain data points belonging to a wrist watch action

# Distributed context modeling

...For instrumented cities?!?..

# Distributed context aware interaction

(Celentano et al. 2002, 2003)

- Problem: interaction in large intelligent environments
- Solution: use agents to structure the problem space
  - Interaction Locus *IL* (3d-space) and the User
  - Agents: Genius loci and User's numen
- Target Scenario: intelligent museum guide

# Distributed context aware interaction

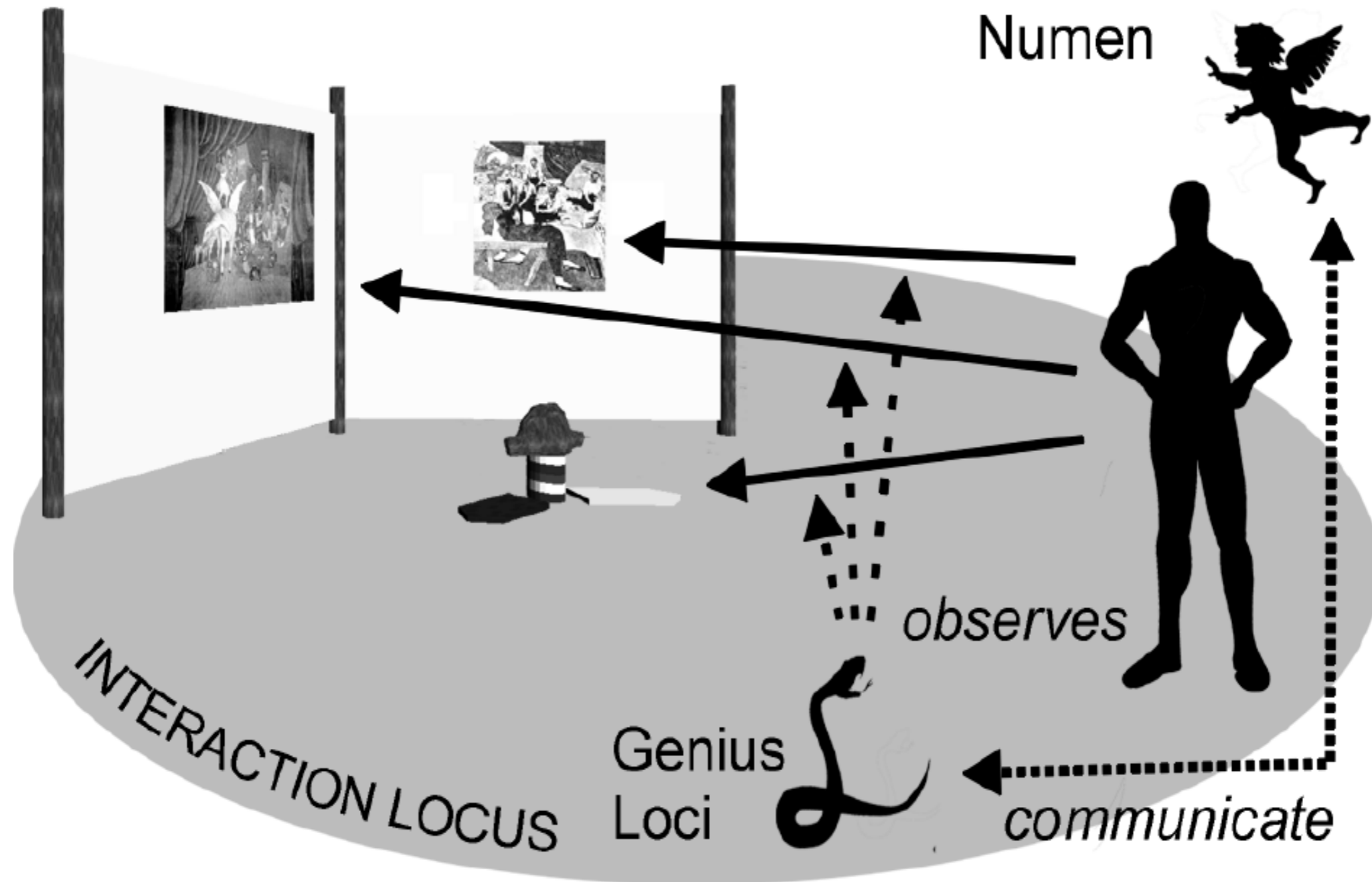
## ■ Genius loci:

- (lat.), Geist des Ortes
- Knows about displays and interaction possibilities of the interaction locus

## ■ User's numen

- Numen (lat.), göttl. Wesen, Gottheit ohne persönl. Gestalt.
- Knows about the interests and profile of the user and his exploration history

# Distributed context aware interaction



# Distributed context aware interaction

- Communication protocol integrates knowledge and is started when the user enters the *IL*
  1. Genius Loci (GL) starts the dialog and contacts the user's numen (UN)
  2. The UN explains the user's interest and the GL adapts the properties of the interaction
  3. The GL learns user preferences and interaction styles
  4. When leaving the IL the GL informs the UN about his inferences



# SW infrastructure wrap-up

- No single all-purpose infrastructure (Ubicomp-OS ;-)
- Some basic structures used in many variations
  - Blackboard architectures
  - Agent-based systems
  - Pipe-and-filter architectures
  - Service-oriented architectures

# Interaction in instrumented environments

# Major interaction paradigms

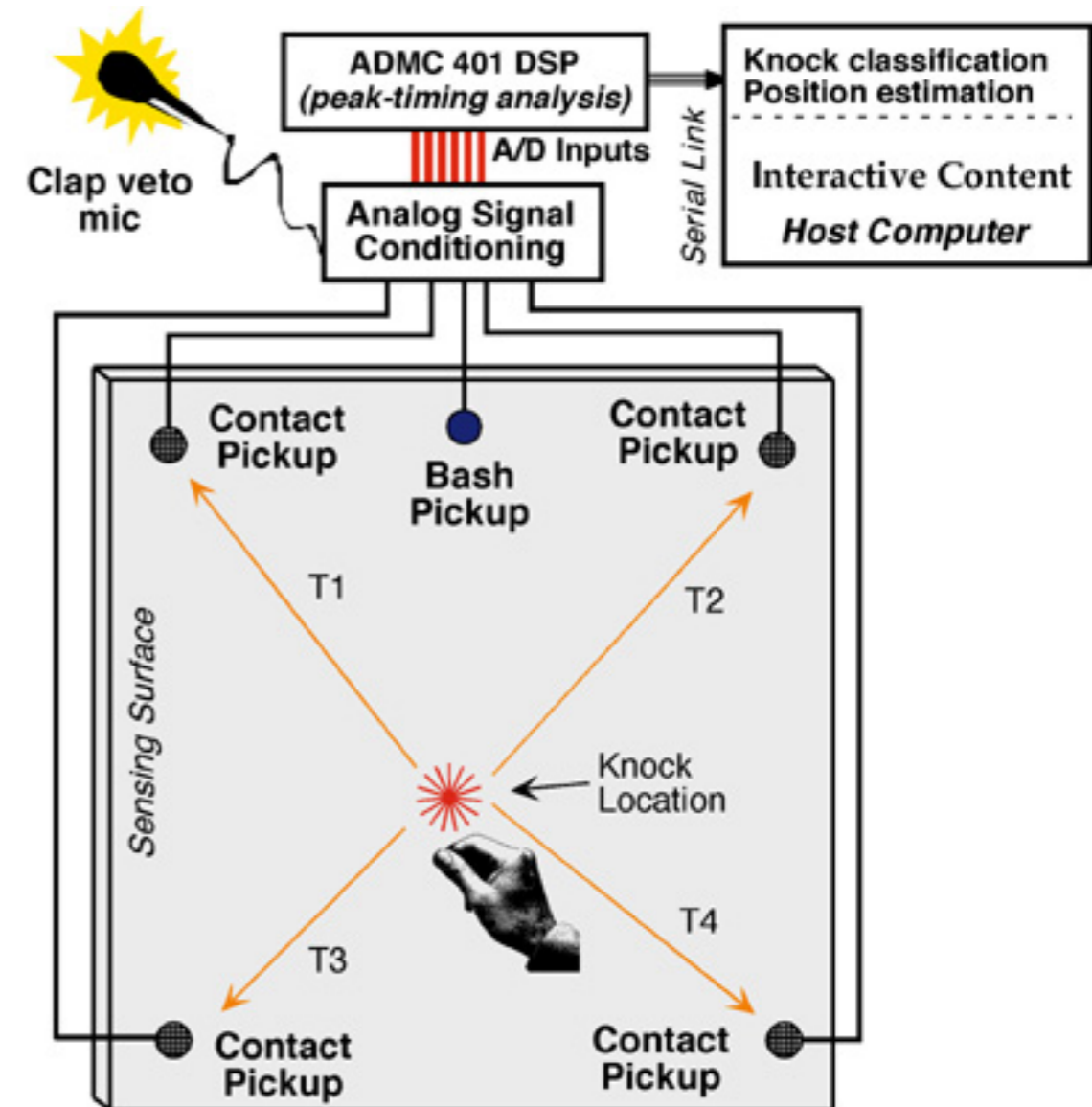
- direct physical interaction
- remote interaction
- implicit interaction
- tangible interaction
- ambient Uis
- interface agents
- interaction models
  - strictly tool-based
  - automation, assisted living
  - proactivity, intelligent agents

# Direct physical interaction

- Touch screens
  - Simple finger buttons (as on ATM machines)
  - Pen-based control of a desktop GUI
  - Also used in fluidum brainstorming demo
- Interactive surfaces
  - Interaction with fingers on everyday surfaces
  - Sensing technology embedded or camera-based interaction (finger recognition)
  - Example: window tap interface

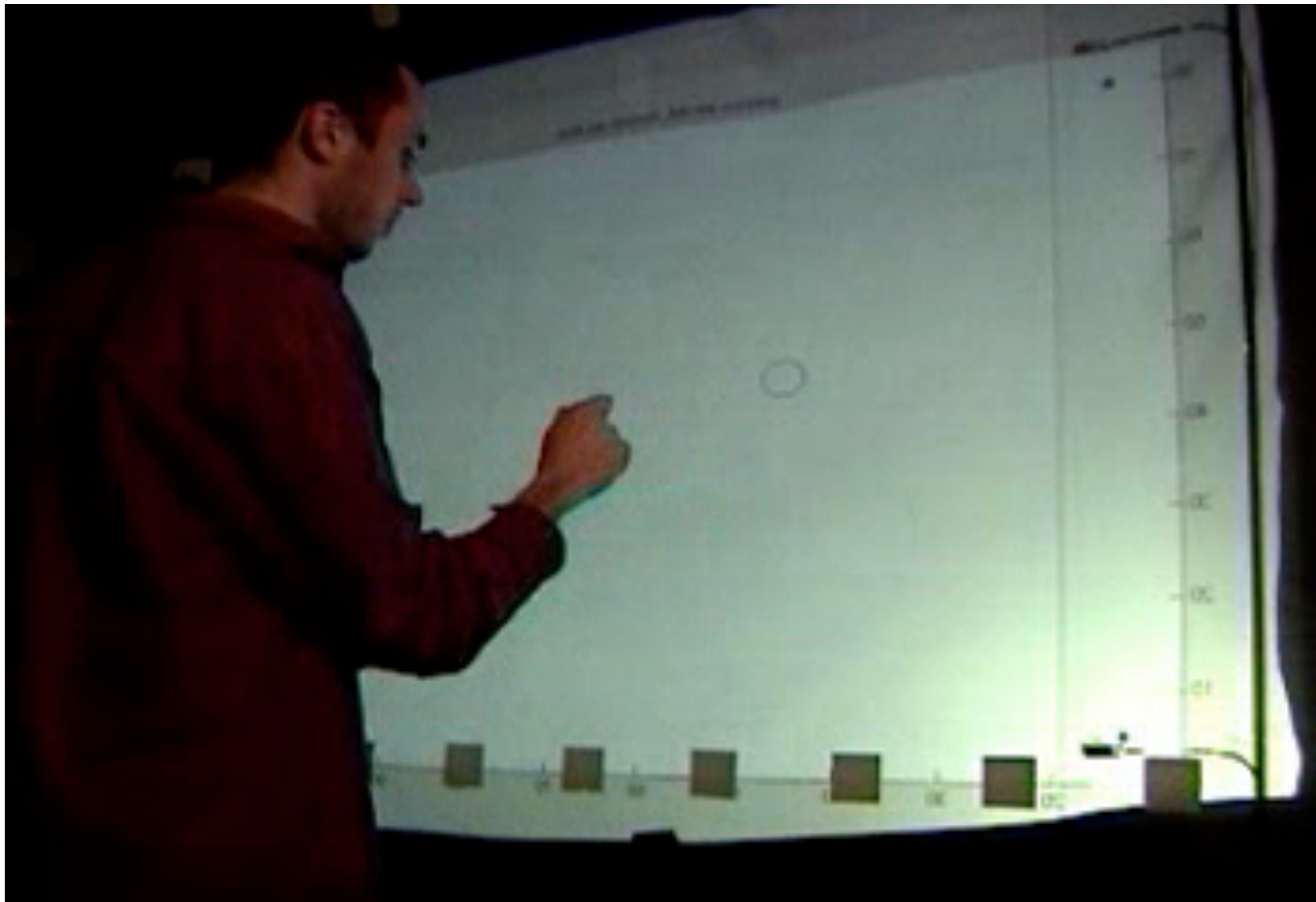
# Example: Window Tap Interface

- locates the position of knocks and taps atop a large sheet of glass.
- piezoelectric pickups
  - located near the sheet's corners
  - record the structural-acoustic wavefront
  - relevant characteristics from these signals,
    - amplitudes,
    - frequency components,
    - differential timings,
  - to estimate the location of the hit
  - simple hardware
  - no special adaptation of the glass pane
  - knock position resolution of about  $s=2$  cm across 1.5 meters of glass



<http://www.media.mit.edu/resenv/Tapper/>

# Example: Window Tap Interface



<http://www.media.mit.edu/resenv/Tapper/>

# Example: Window Tap Interface



- <http://www.media.mit.edu/resenv/Tapper/>

# DMI: what is „Direct“ ??

- Definition by Shneiderman:
  - continuous representation of the object of interest,
  - physical actions or labeled button presses instead of complex syntax,
  - rapid incremental reversible operations whose impact on the object of interest is immediately visible.
- Stems from a time when command line interfaces were the rule (1983)
- Mouse interaction = direct interaction?
- Touch screens = more direct?



# Remote interaction

- Laser pointer interaction
  - E.g., [Olsen & Nielsen, CHI 2001](#)
- Gesture interaction
  - E.g., [Vogel & Balakrishnan, UIST 2005](#)
- Mobile phone interaction
  - Markers
  - Motion
- Pointing, scanning, touching
  - E.g., [Välkkynen et al. PI03](#)

# Laser Pointer Interaction

[[Olsen & Nielsen, CHI 2001](#)]

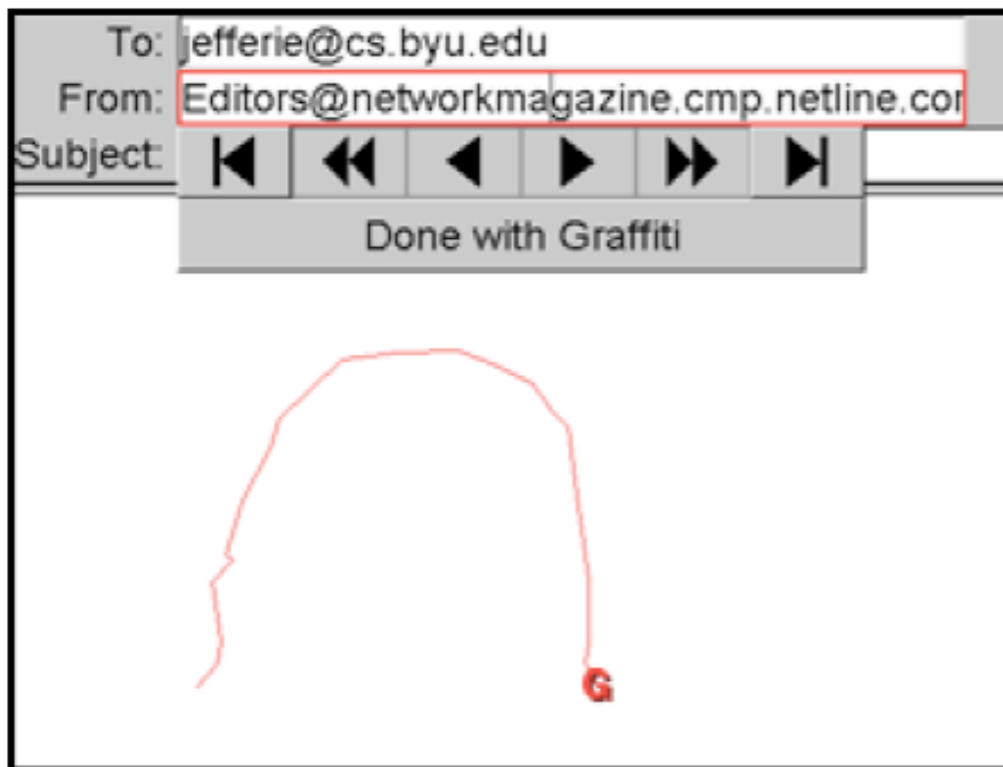
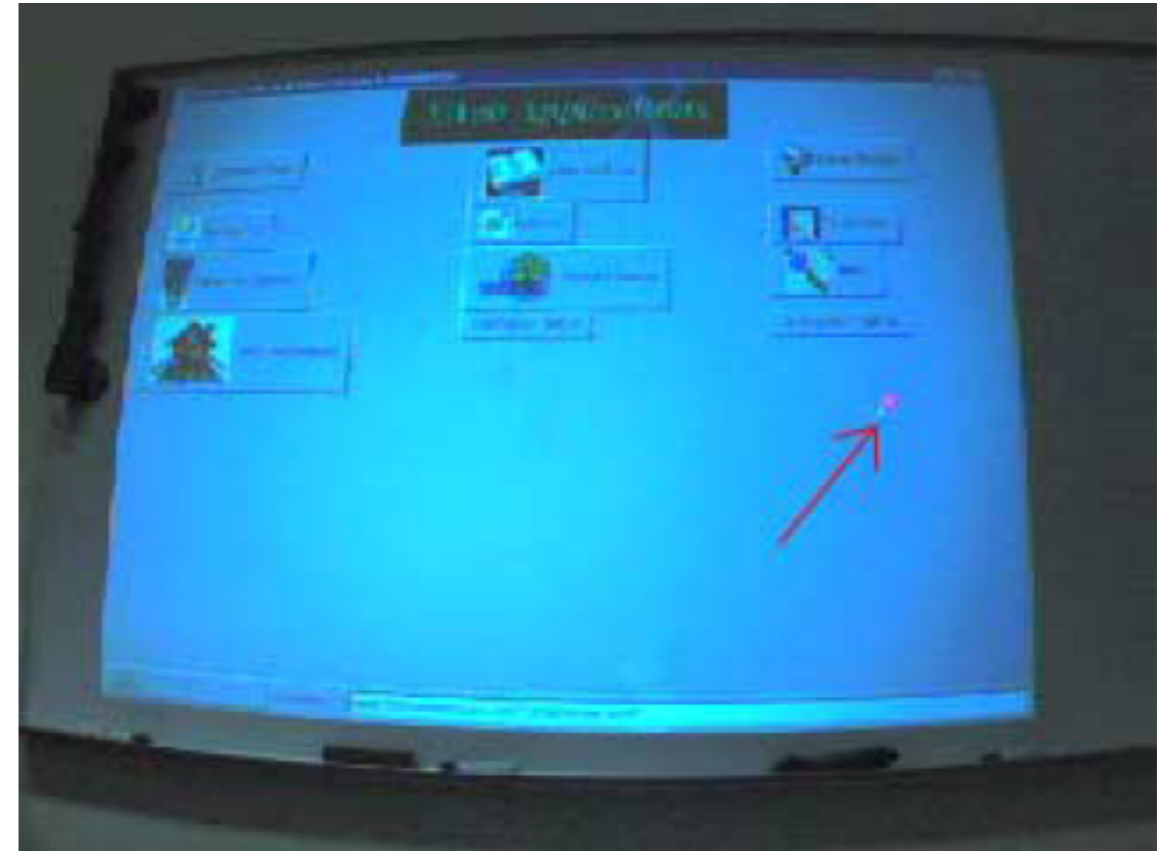
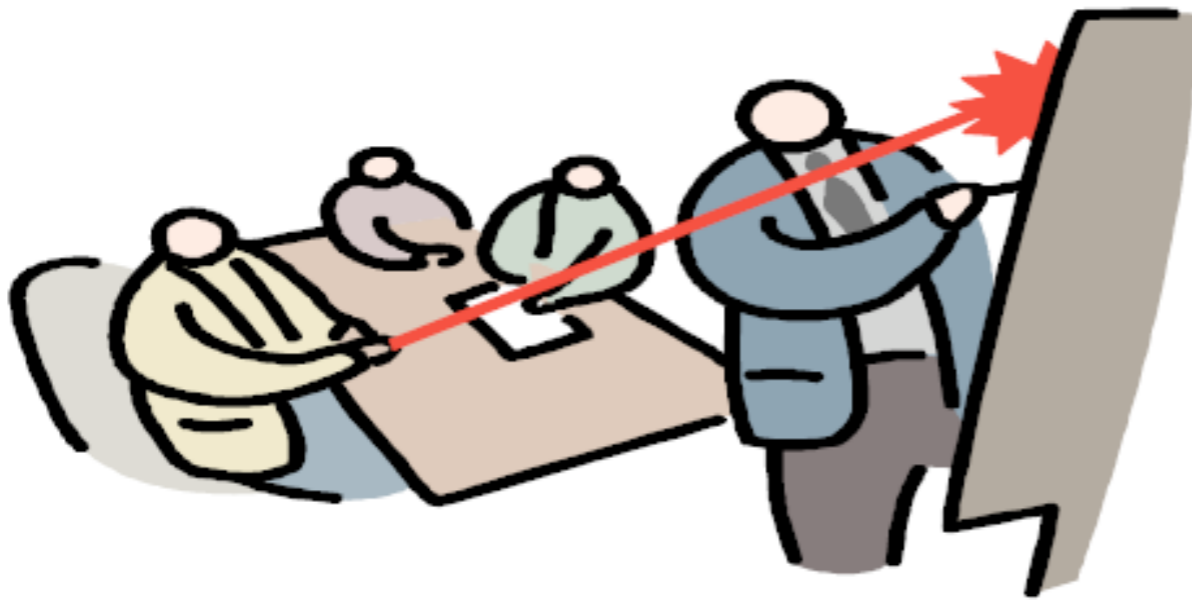
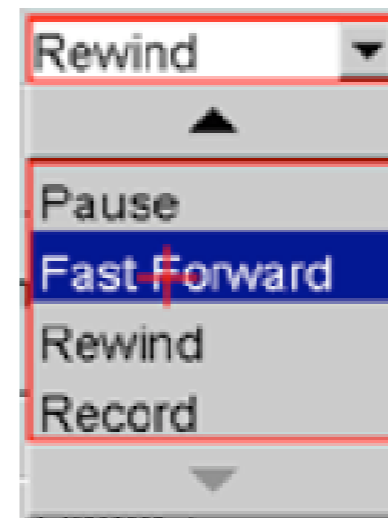


Figure 9 - Text Entry



# Distant Freehand Pointing and Clicking [\[Vogel & Balakrishnan, UIST 2005\]](#)

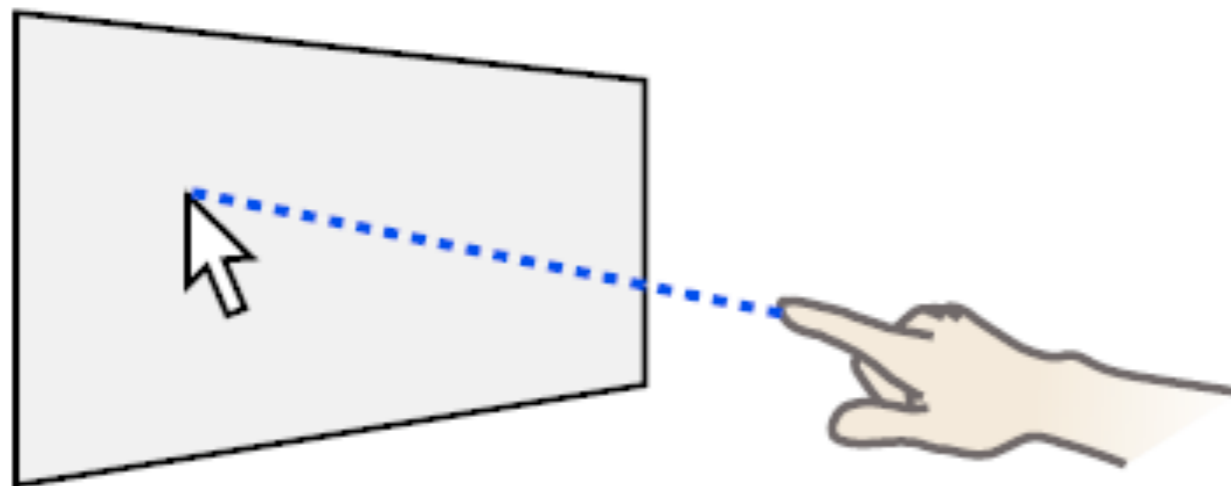
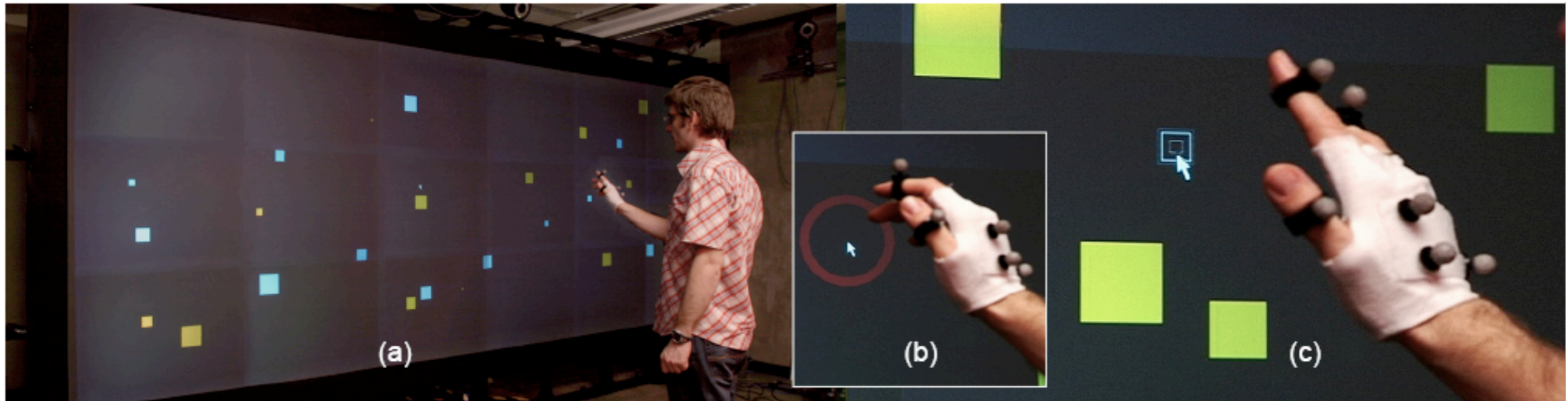


Figure 5. RayCasting. A ray extends from the tip of the finger and the cursor is positioned where it intersects with the large display surface.

# Kickass Kung Fu ([Perttu Hämäläinen](http://www.kickasskungfu.net/))

<http://www.kickasskungfu.net/>



# Implicit interaction

- Interaction is not done explicitly
  - Just as a side effect of other actions
  - Might be unnoticed by user
- Example RFID shopping assistant
- Example media cup

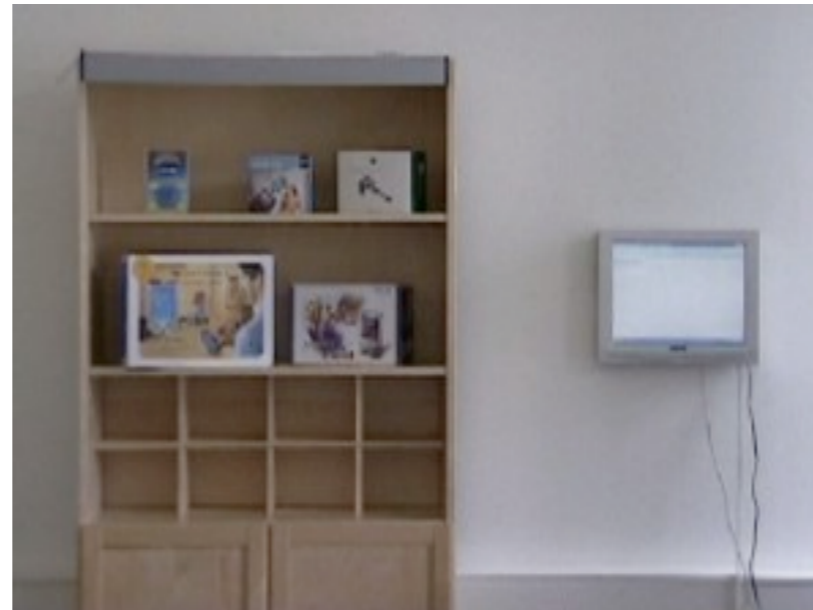
# Mediacup

- Cup sensing temperature, weight and movement
- Location of cups detected
- Detected interaction:
  - Presence of multiple people in a room → mark room as occupied



# RFID shopping assistant

<http://www.misch.net/ssa/>



- Tagged objects in a shopping shelf
- Antenna registers when object is taken out
- Antenna in cart registers when object is put in cart
- Detected interactions:
  - Inspecting an object → give additional info
  - Inspecting several objects → compare
  - Decision to buy an object → advertise additional objects

# Tangible User Interfaces

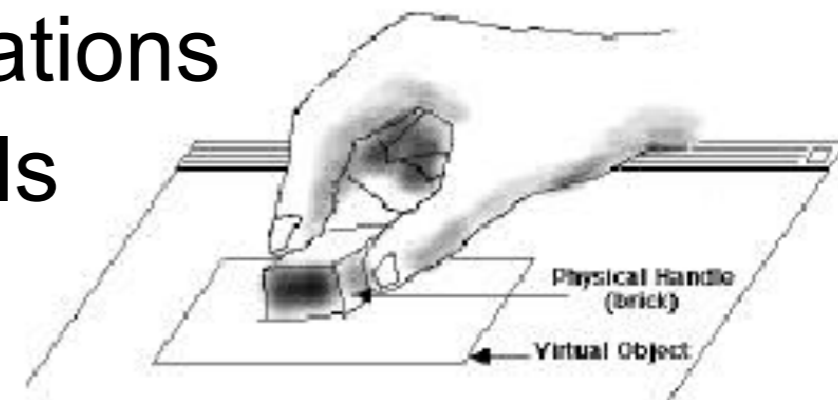
General purpose TUI frameworks



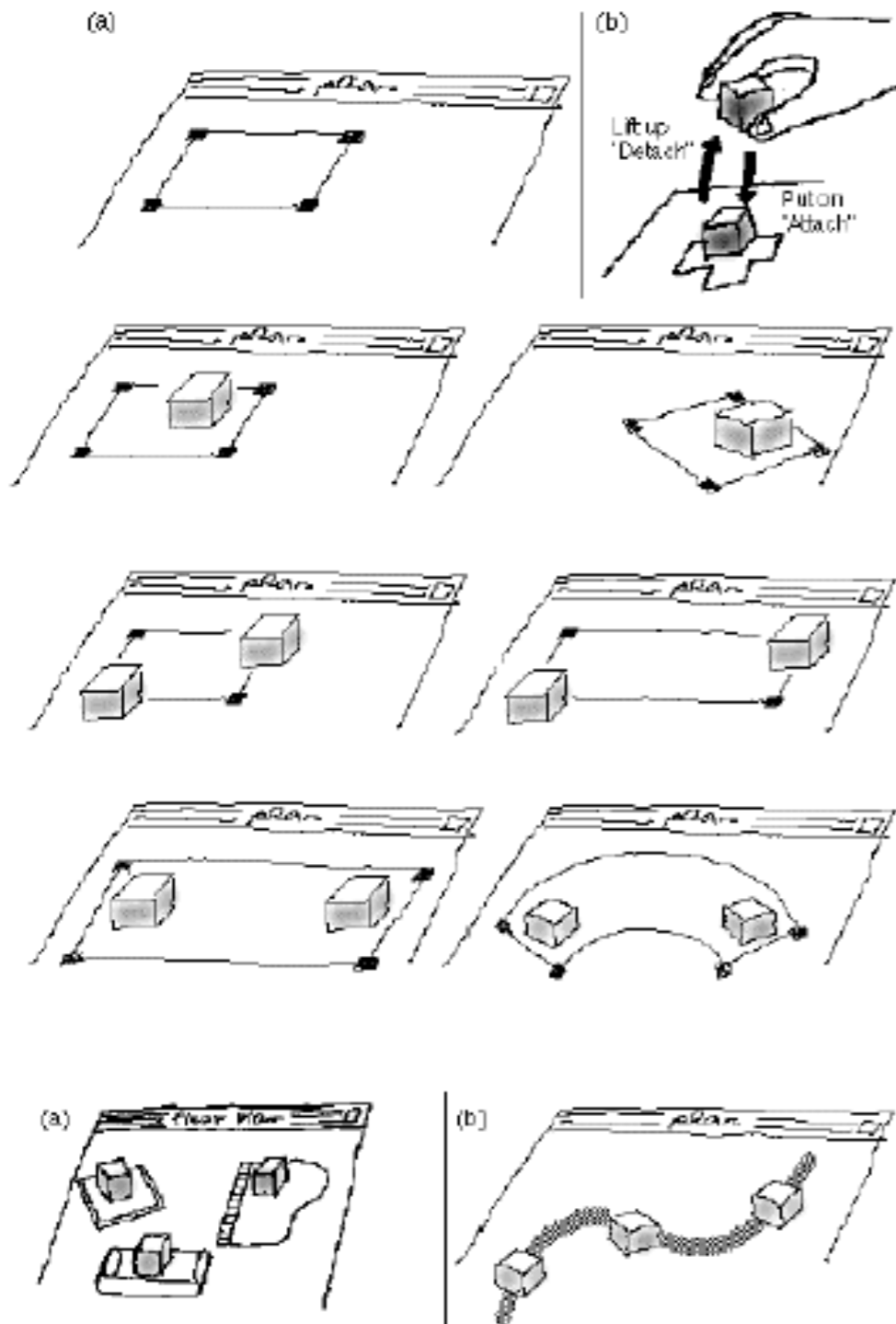
# Bricks: Graspable User Interfaces

(Fitzmaurice, Ishii, Buxton, CHI 95)

- specialized, context sensitive input devices
- interface elements more "direct" and more "manipulable" by using physical artifacts
- parallel input specification by the user
  - improving the expressiveness or the communication capacity with the computer
- encourages two handed interactions
- leverages our everyday skills of prehensile behaviors for physical object manipulations
- externalizes internal computer representations
- takes advantage of spatial reasoning skills
- affords multi-person, collaborative use



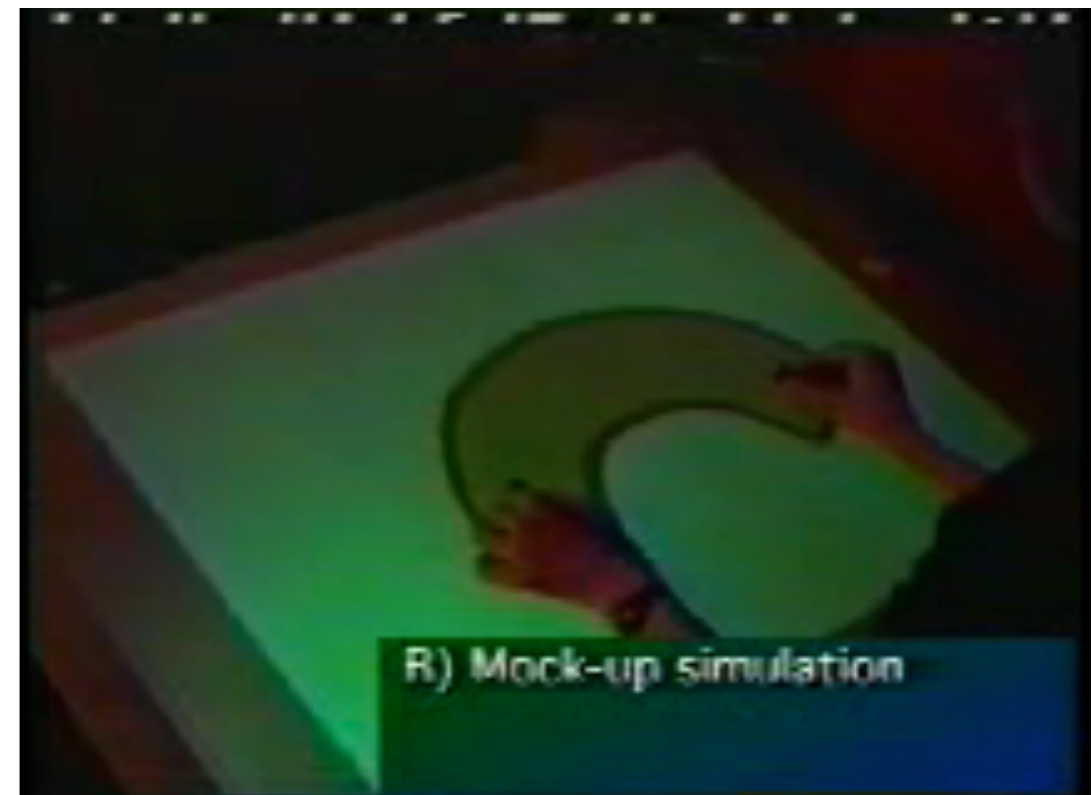
# Bricks: basic operations



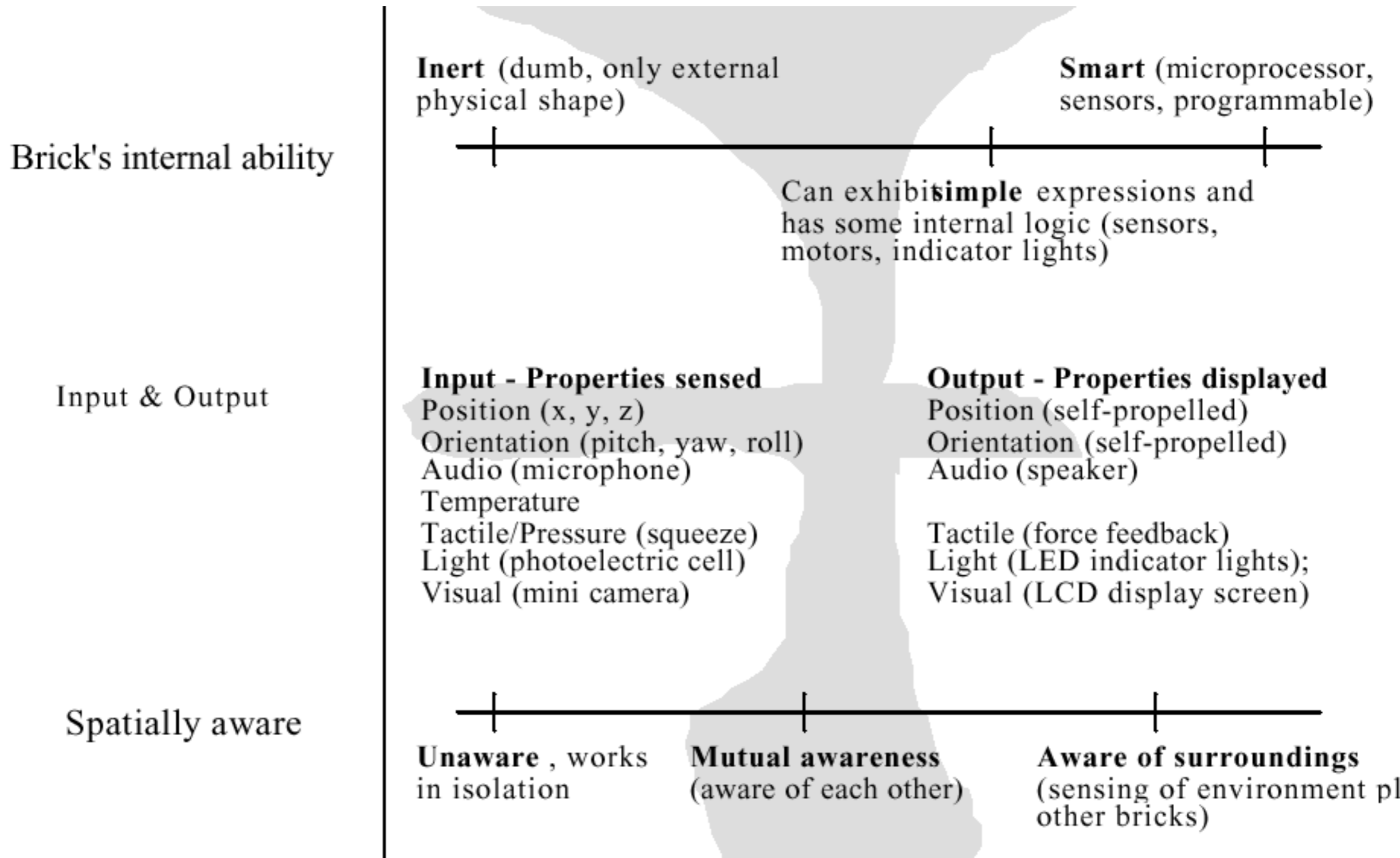
- Select an object
- Move and rotate
- Scale and stretch
- Bend and deform
- Floor planning, curve drawing

# Bricks application: GraspDraw

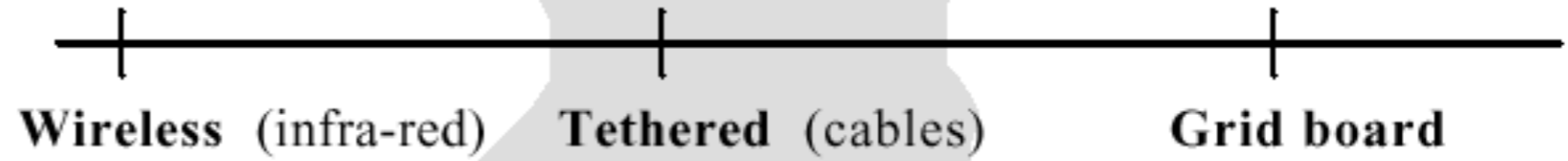
- Drawing application
- On active desk
  - Rear-projection display
  - Transparent digit. Tablet
  - Magnetic tracker for bricks
- Two bricks for input
  - „Anchor“ and „actuator“



# Bricks: Design Space



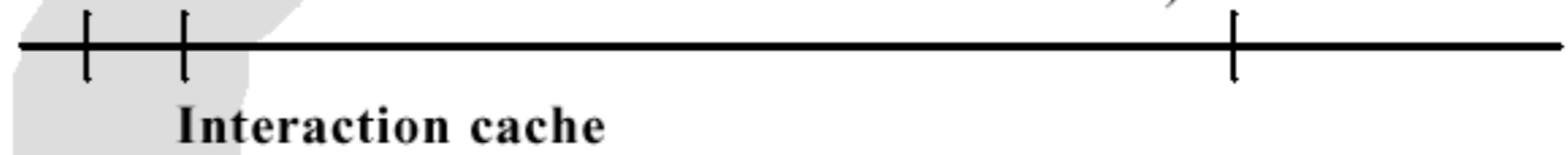
Communication  
(inter-brick and to host)



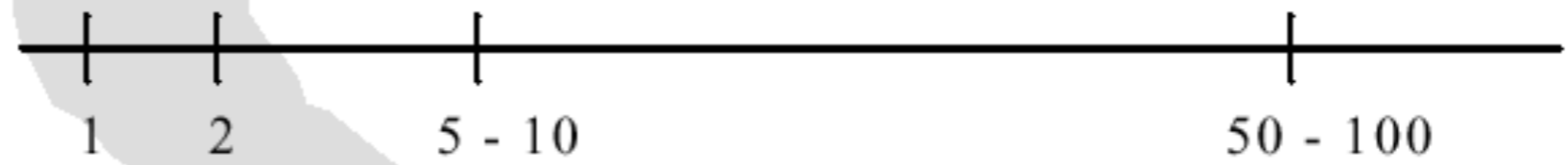
**Quick**, gestures, fraction of seconds (specify parameter, initiate process)

**Long term**, (days, months, years between interactions; archives)

Interaction time span



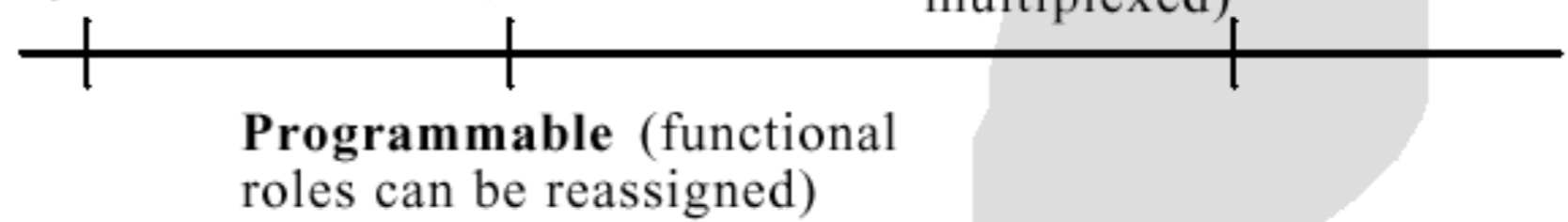
Bricks in use at same time



**Permanent** (each brick assigned one function)

**Transient** (rapid reassignment; time multiplexed or space multiplexed)

Function assignment

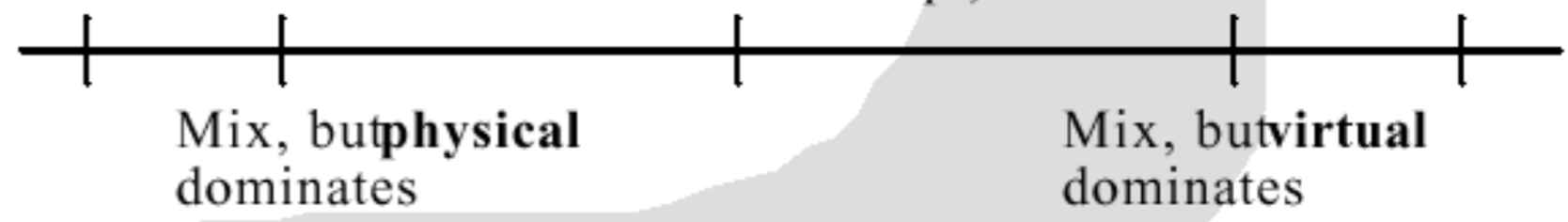


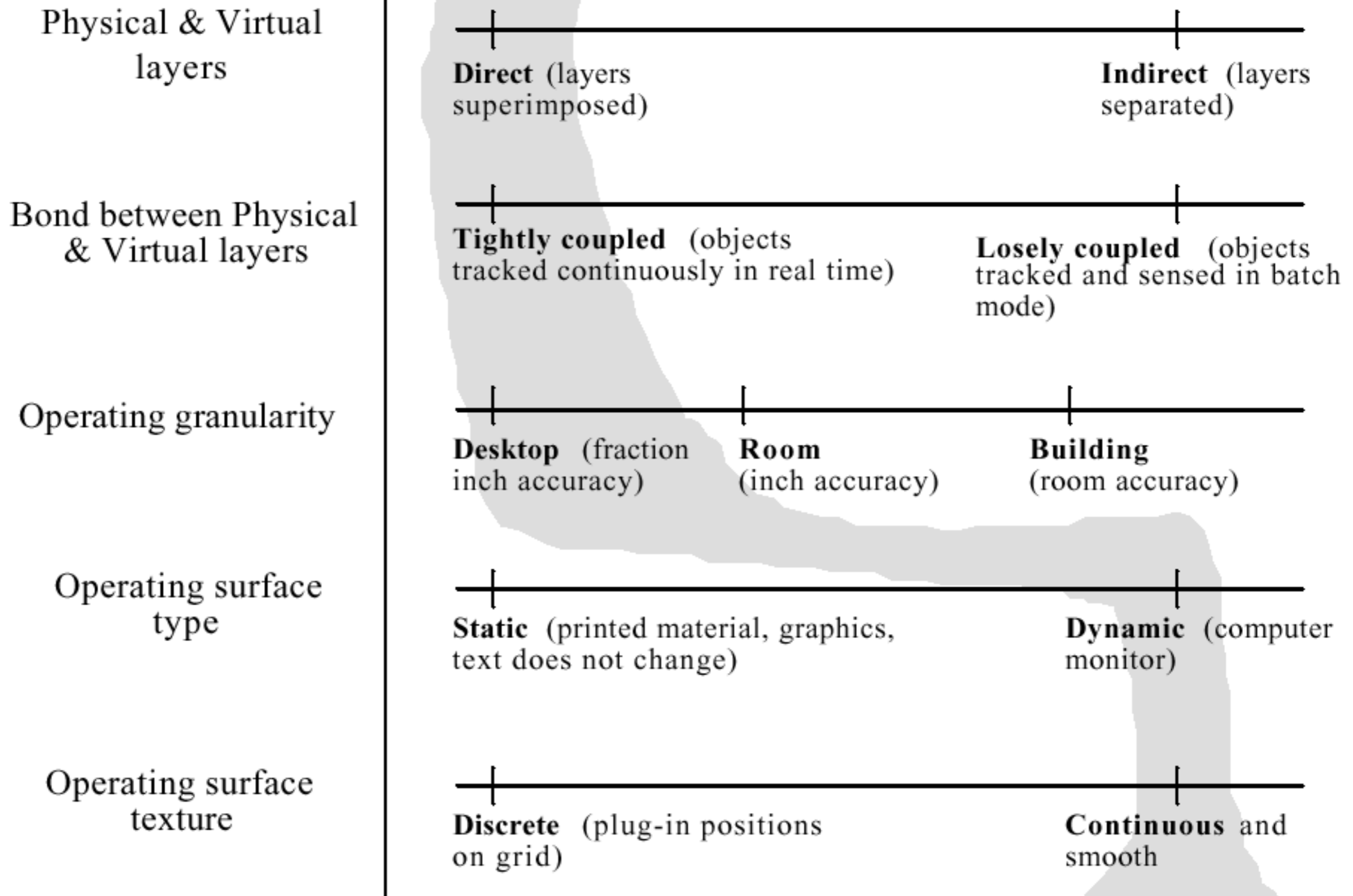
All **physical** artifacts

**Balanced mix** (Equal, Complimentary or Combinatoric rep.)

All **virtual** artifacts

Interaction representations





# Triangles

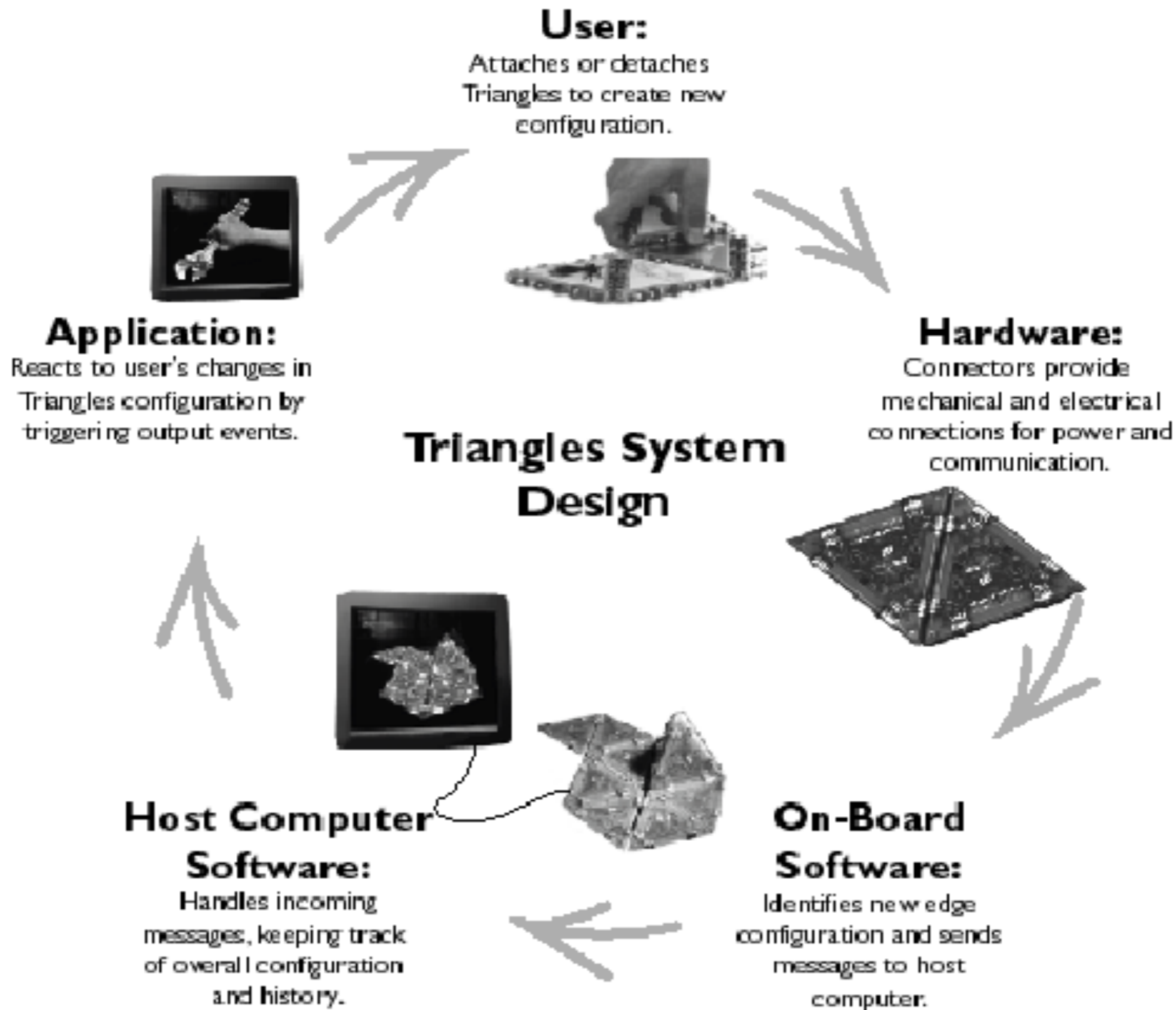
(Gorbet, Orth, Ishii, CHI 98)

- Set of identical, flat plastic triangles
  - Each with a processor and a unique ID
  - Magnetic edge connectors
- Can be rearranged in 2D and 3D
  - Keep track of their connections
  - Transmit their configuration to a PC
- Building blocks for topographies
  - Immediate physical interaction
  - Spatial language



*photo:  
Webb Chappell*

# Triangles: System overview





# Triangles: Example applications



Fig. 8: The *Cinderella 2000* Triangles

Non-linear storytelling

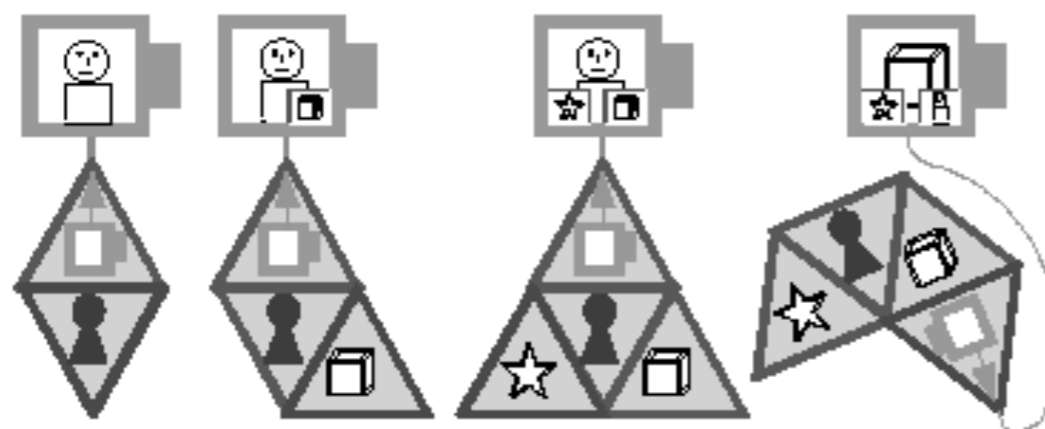
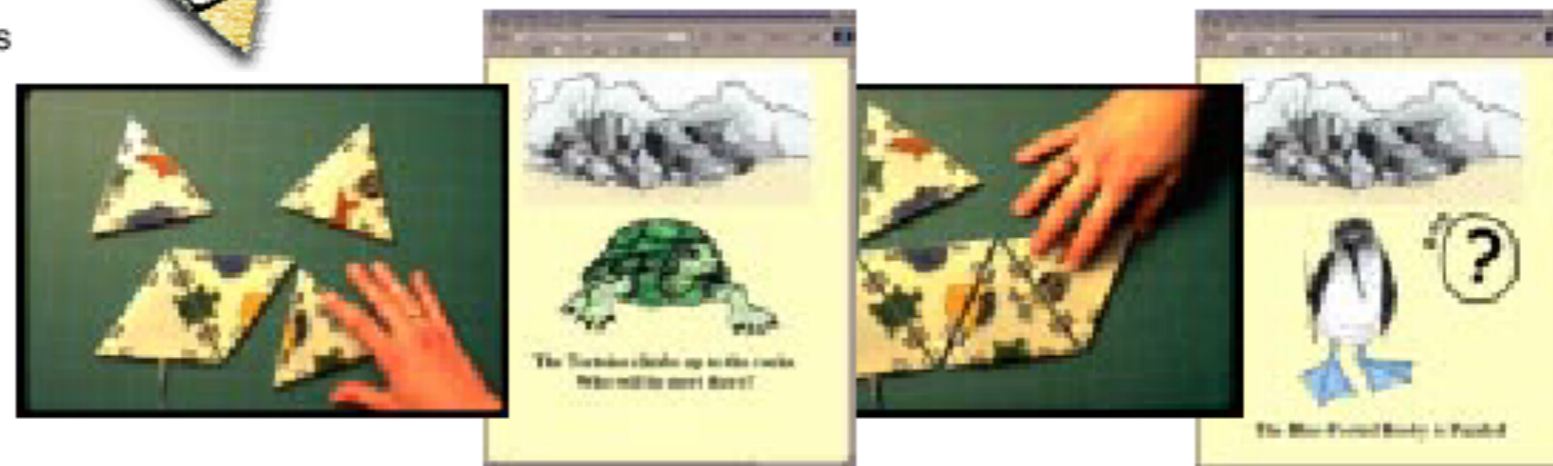


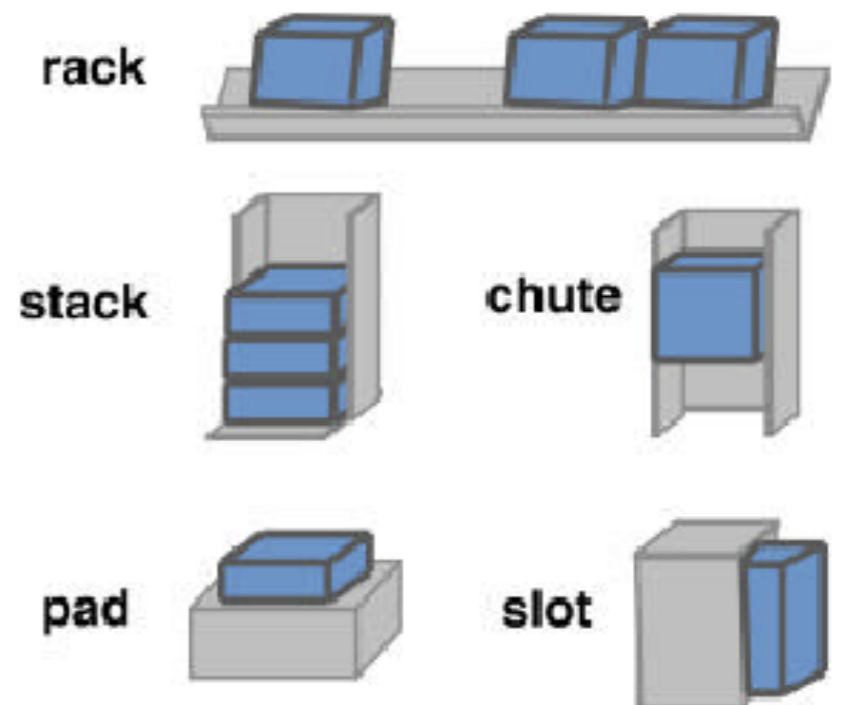
Fig. 9: *TriMediaManager*

Media Management

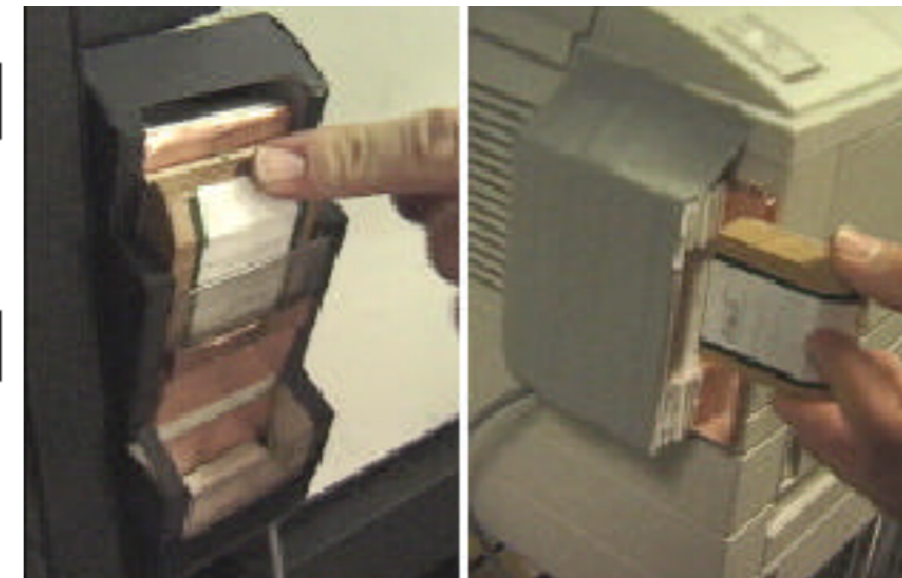
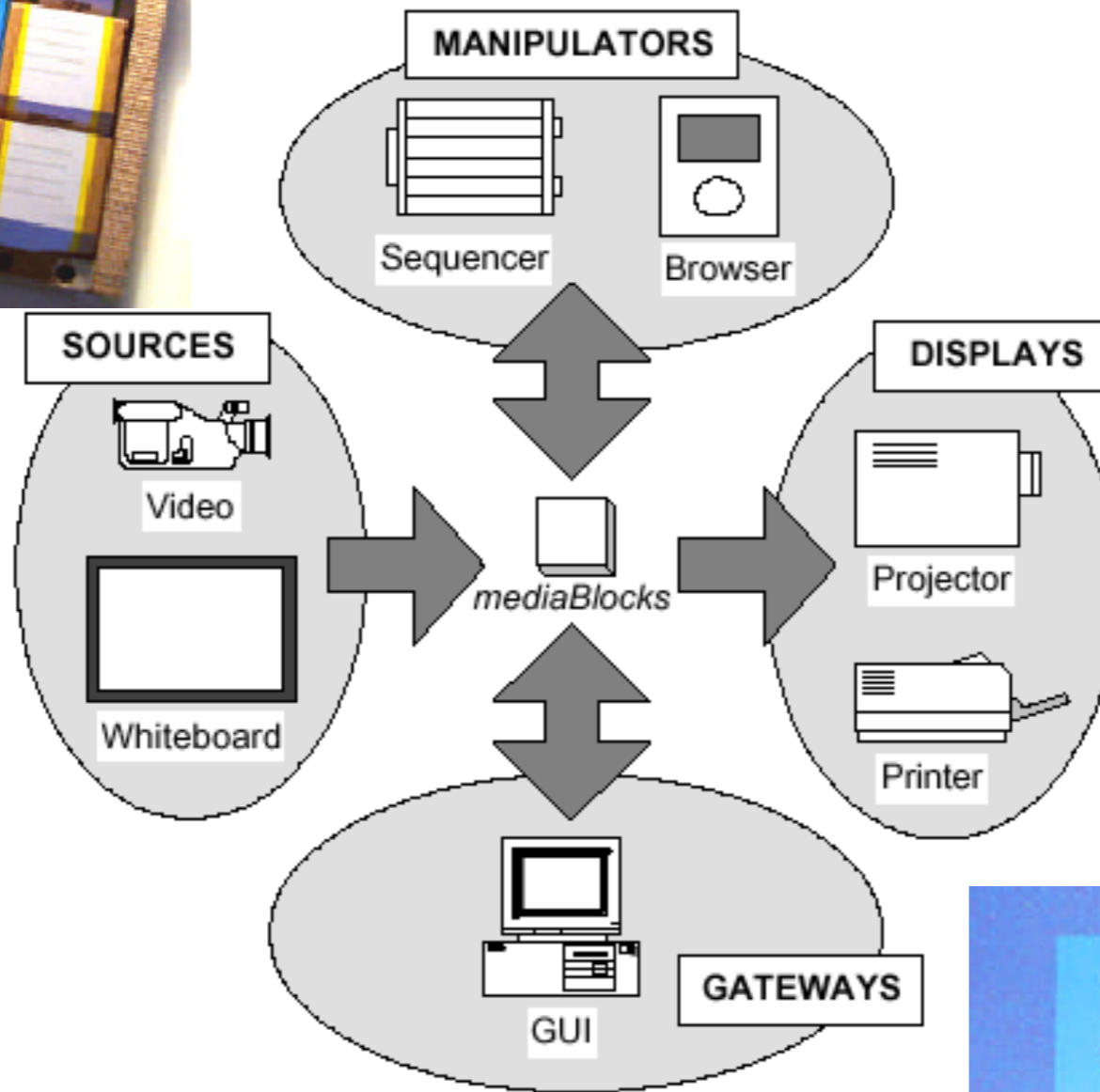
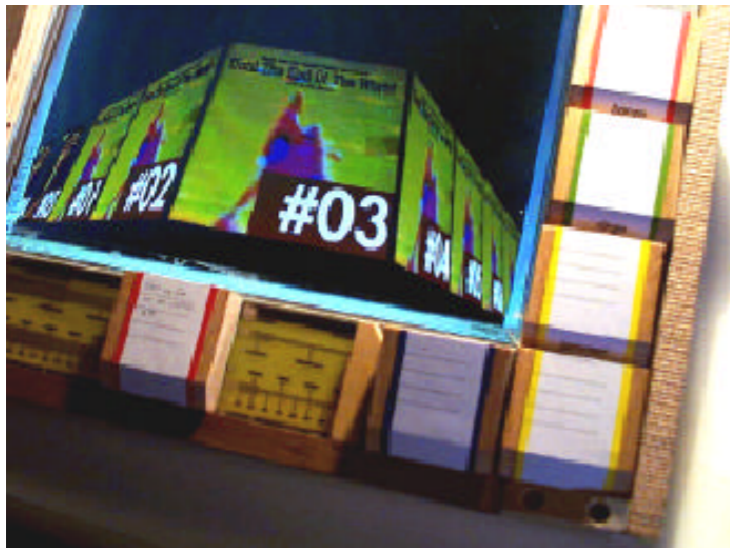
# MediaBlocks

(Ullmer, Ishii, Glas, SIGGRAPH 98)

- Physical objects representing digital information: **phicons**
- No actual information stored on the blocks
- Various containers with different physical constraints



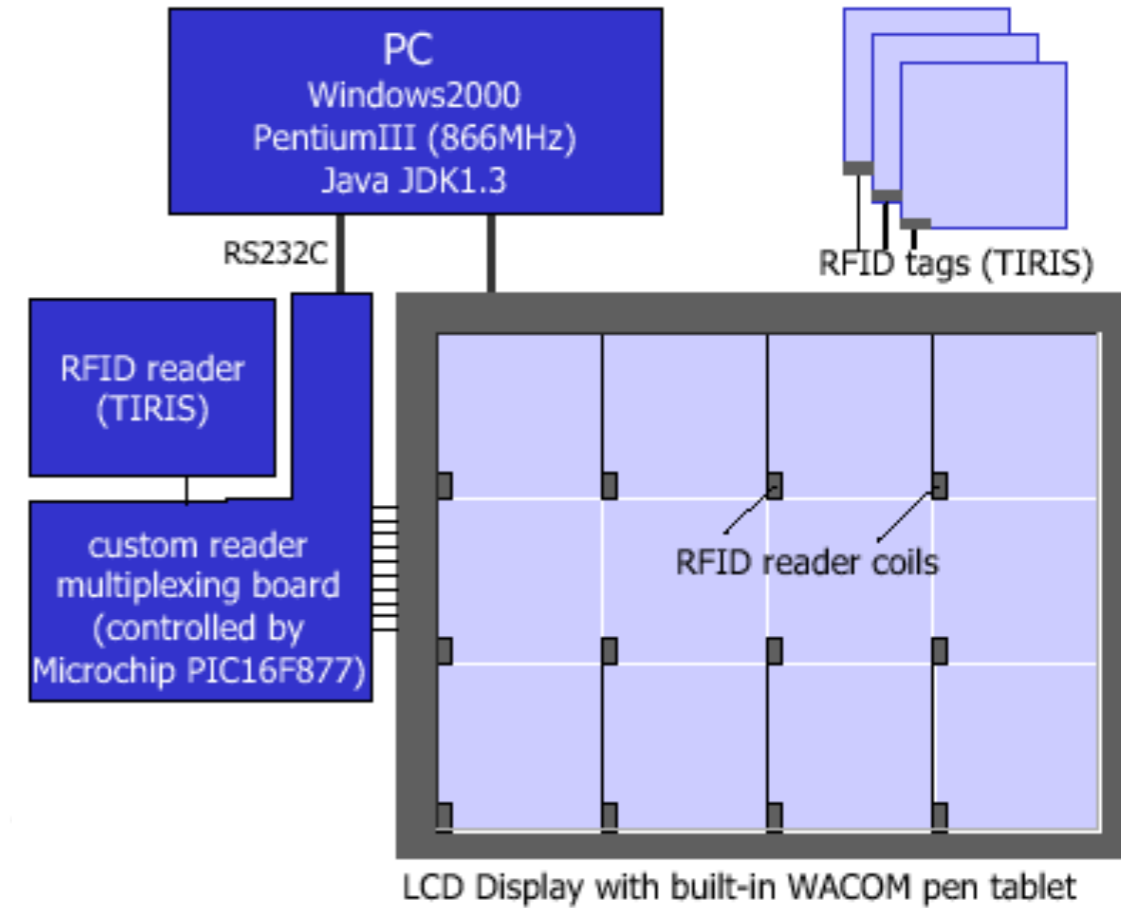
# MediaBlocks (contd.)



# DataTiles

(Rekimoto, Ullmer, Oba, CHI 01)

- Transparent plastic tiles
  - On a flat panel screen
  - Sensed by RFID tags
  - Provide grooves for pen
  - Can be spatially arranged
- Different tile types
  - Application tile
  - Container tile
  - Portal tile
  - Parameter tile



# DataTiles (contd.)

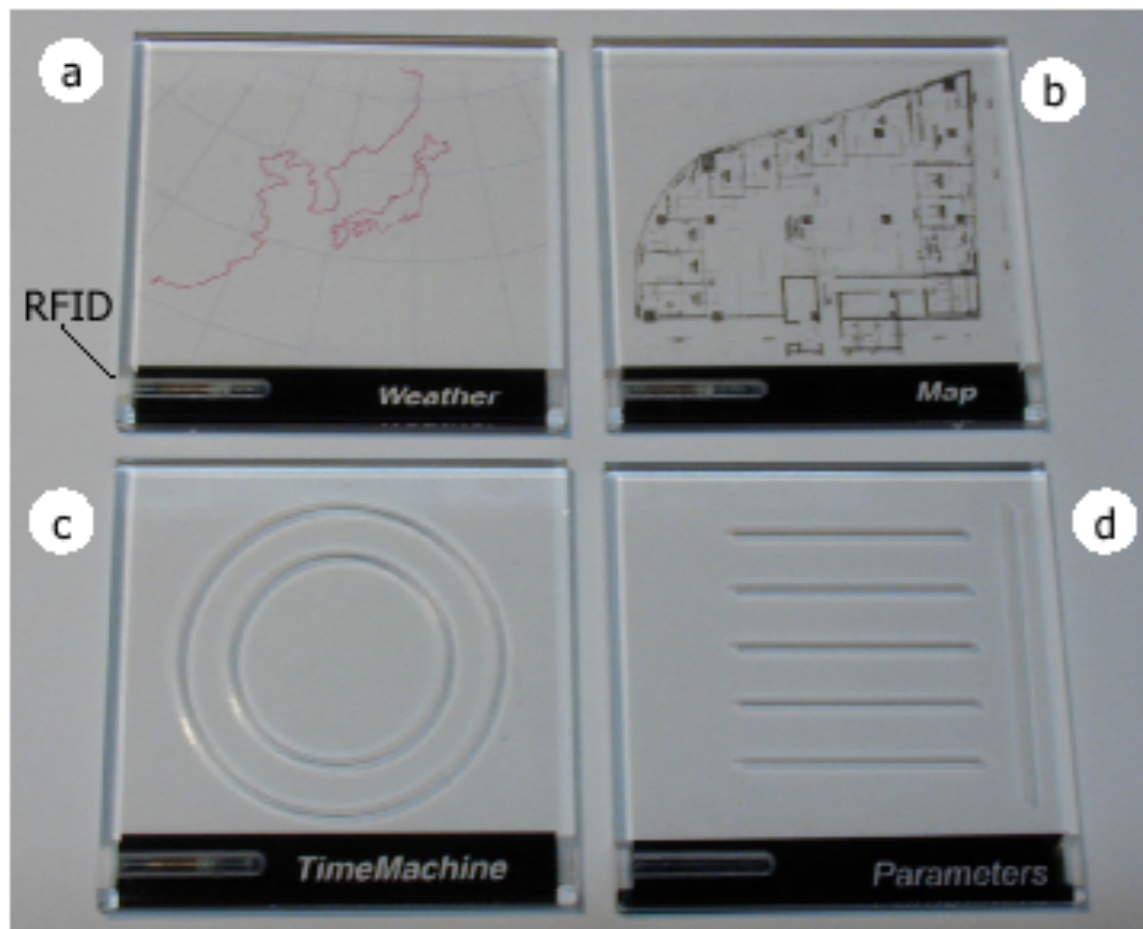


Figure 3: Tile examples. (a) and (b): partially printed tiles, (c) and (d) tiles with “grooves”.

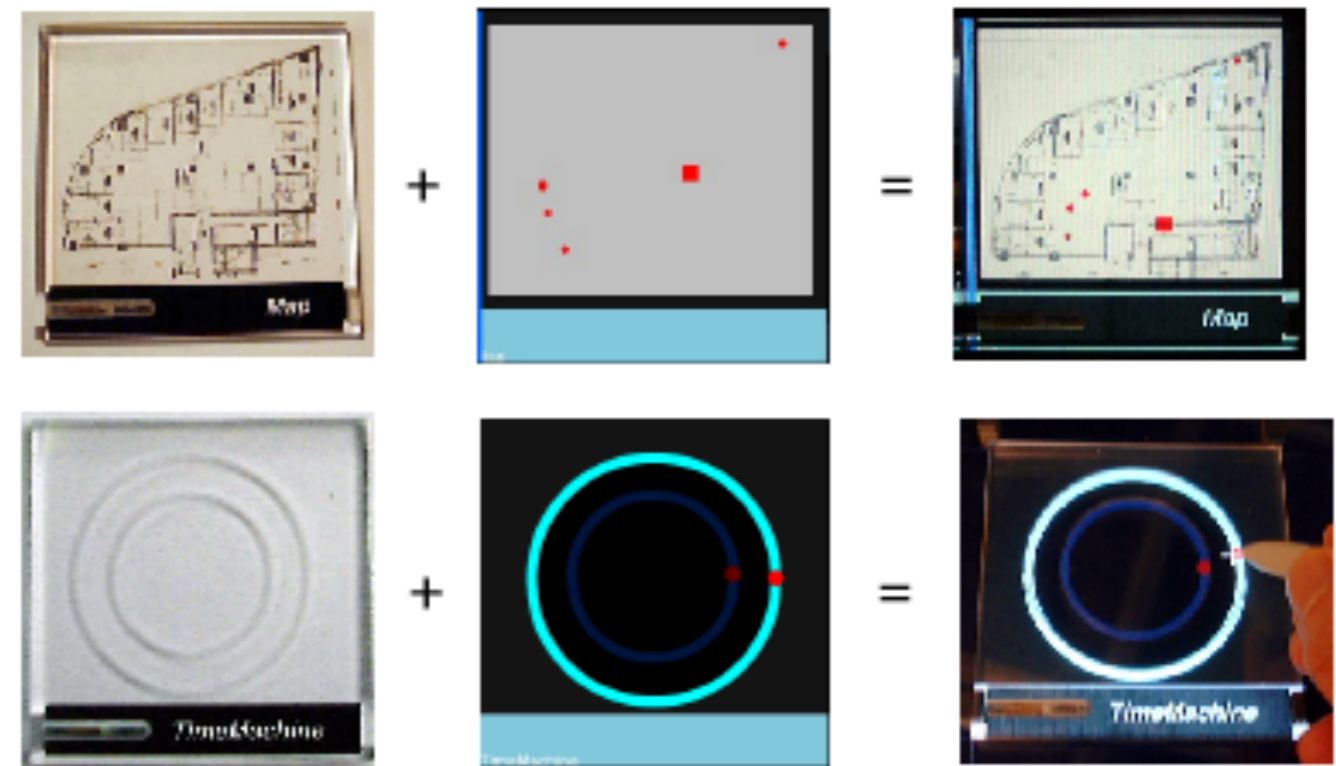
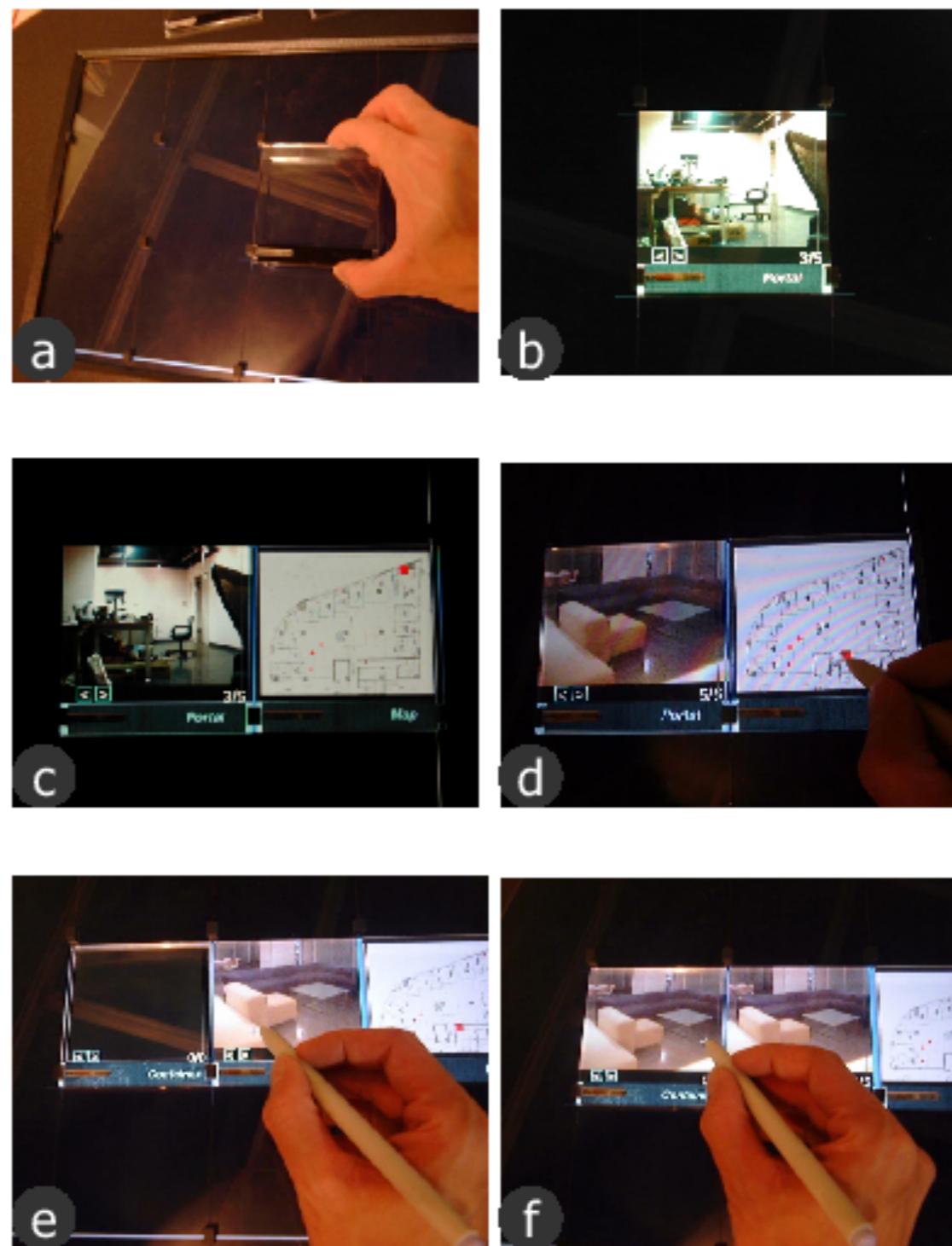
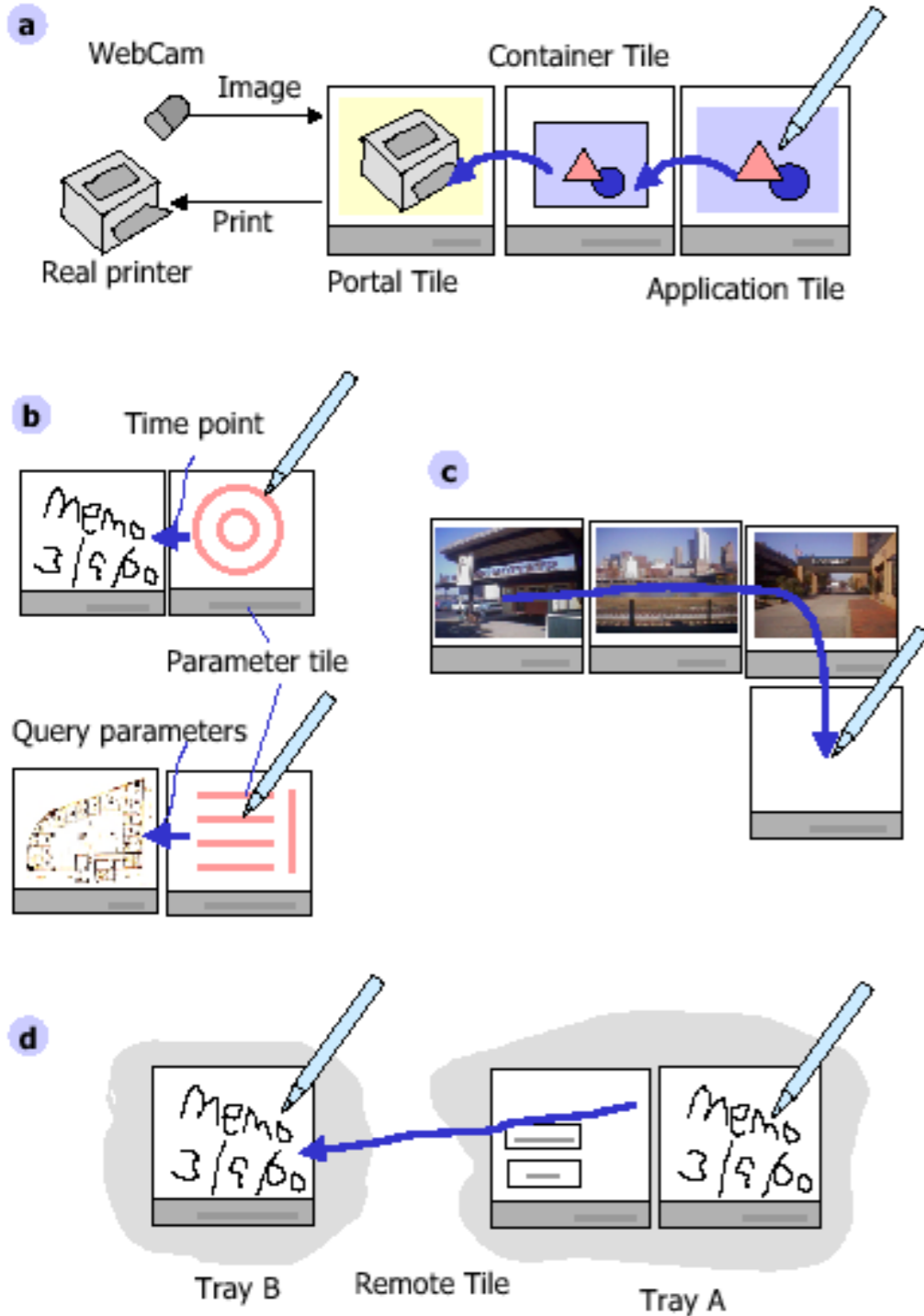


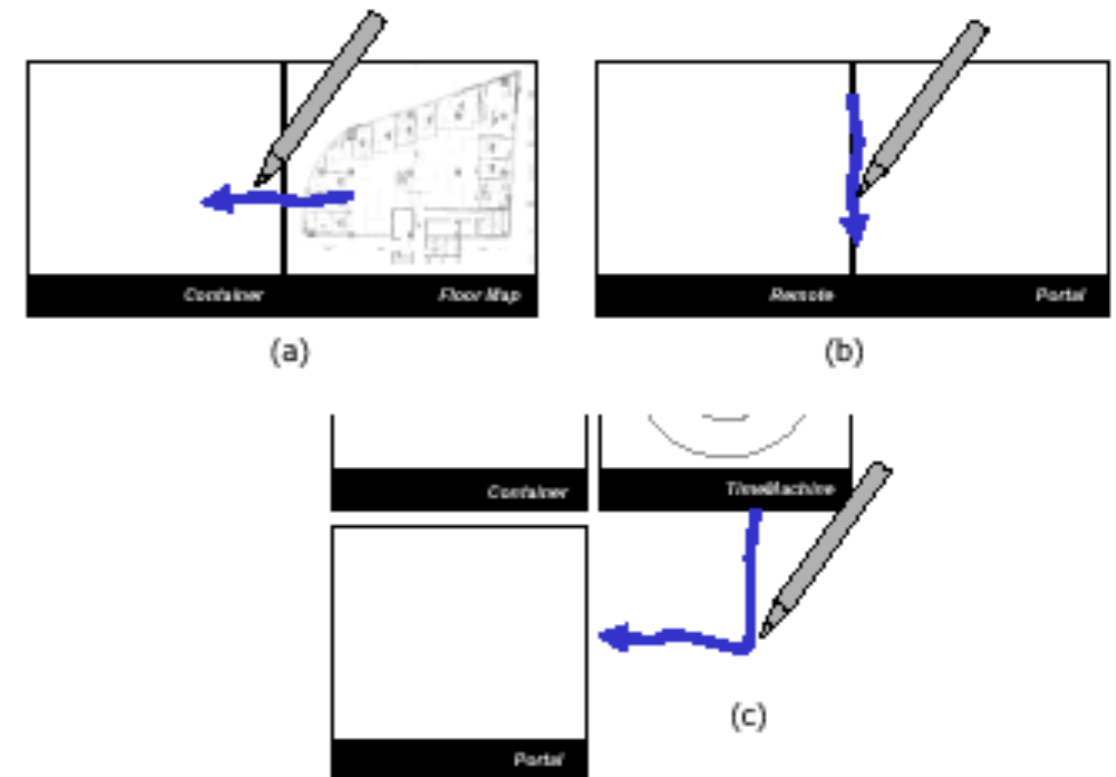
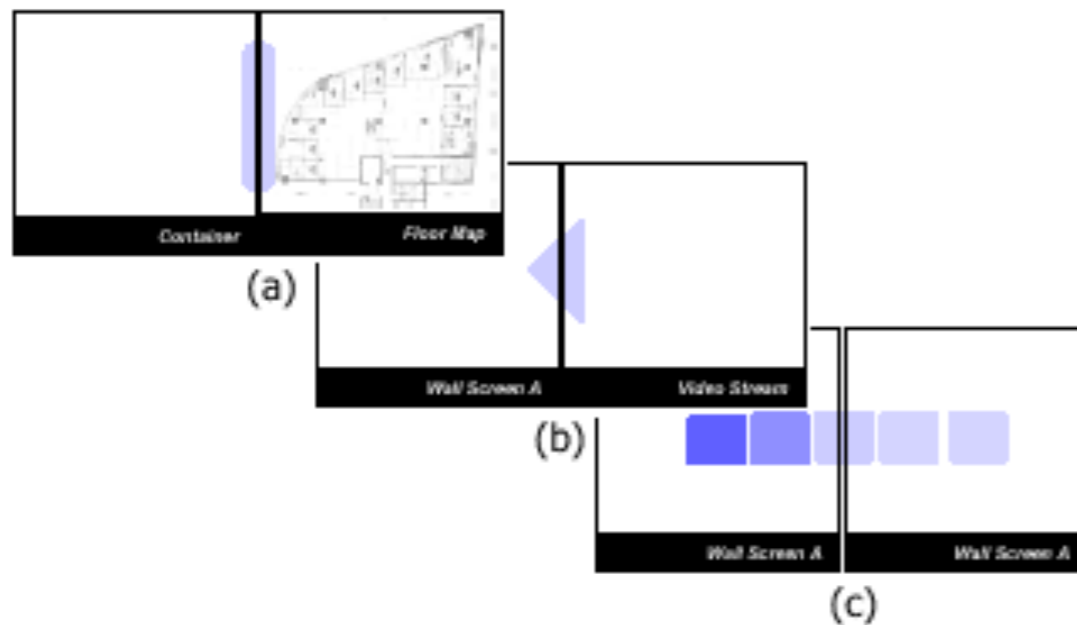
Figure 4: Combination of physical tiles and graphical information. Above: high-resolution printed information can be augmented by displayed graphics. Below: combination of physical grooves and graphical information creates a GUI widget with passive haptics.



**Figure 5: Examples of tiles and tile combinations.** (a) An image from an application tile (right) is stored in a container tile (middle), and then transmitted to the portal tile. The portal tile represents a real world object (a printer in this example). (b) Parameter tiles can be used to specify various types of parameters. (c) Concatenates three video clips and stores item in a container tile. (d) Remote tiles are used to connect distributed tile trays. In this example, a shared drawing environment has been constructed.

**Figure 6: Examples of tile combination:** (a) When a user places a portal tile on the tray, (b) an associated webcam image appears on the tile. (c) Then the user places a map tile, and the map displays locations of webcams. (d) The user clicks on a spot on the map to select another webcam. (e, f) Then the user makes an inter-tile gesture (from portal tile to the container tile) to store a snapshot image in the container tile.

# DataTiles (contd.)



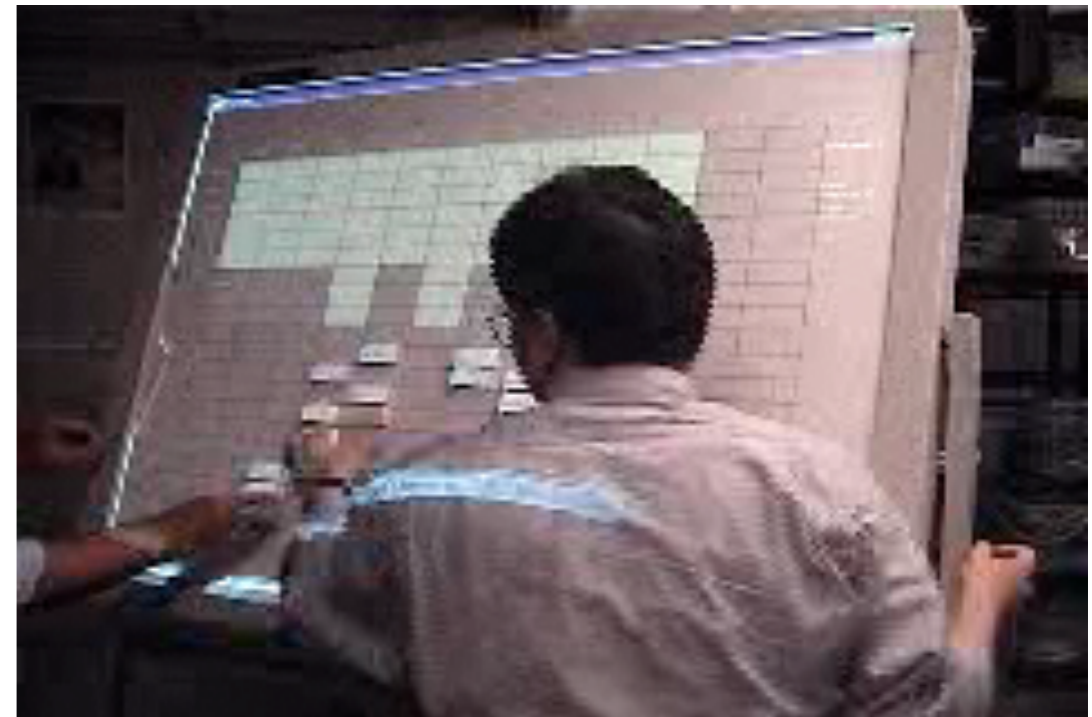
**Figure 8: Several visual feedback approaches for indicating connection types. (a) one-way discrete data transmission from right to left, (b) one-way continuous data transmission, and (c) bi-directional continuous connection using animations.**

**Figure 9: Inter-tile gestures by a pen to control a data connection between two adjacent tiles. (a) triggers a discrete data transmission, (b) suspends a continuous data transmission, and (c) connects two disjoint tiles. (Note: During these operations, the pen tip must be sufficiently close to the tile surfaces to be sensed, but need not touch them.)**

# SenseBoard

(Jacob, Ishii, Pangaro, Patten, CHI 02)

- TUI for organizing information on a grid
- Combines physical manipulation with a computer
  - Physically: arranging cards
  - Computer: arranging icons
- Get the best from both worlds
- Example: organize conference papers into sessions
- Other tasks: arrange songs in a playlist, newspaper articles, slides for a talk, ideas from a brainstorming, emails, bookmarks, notes,...

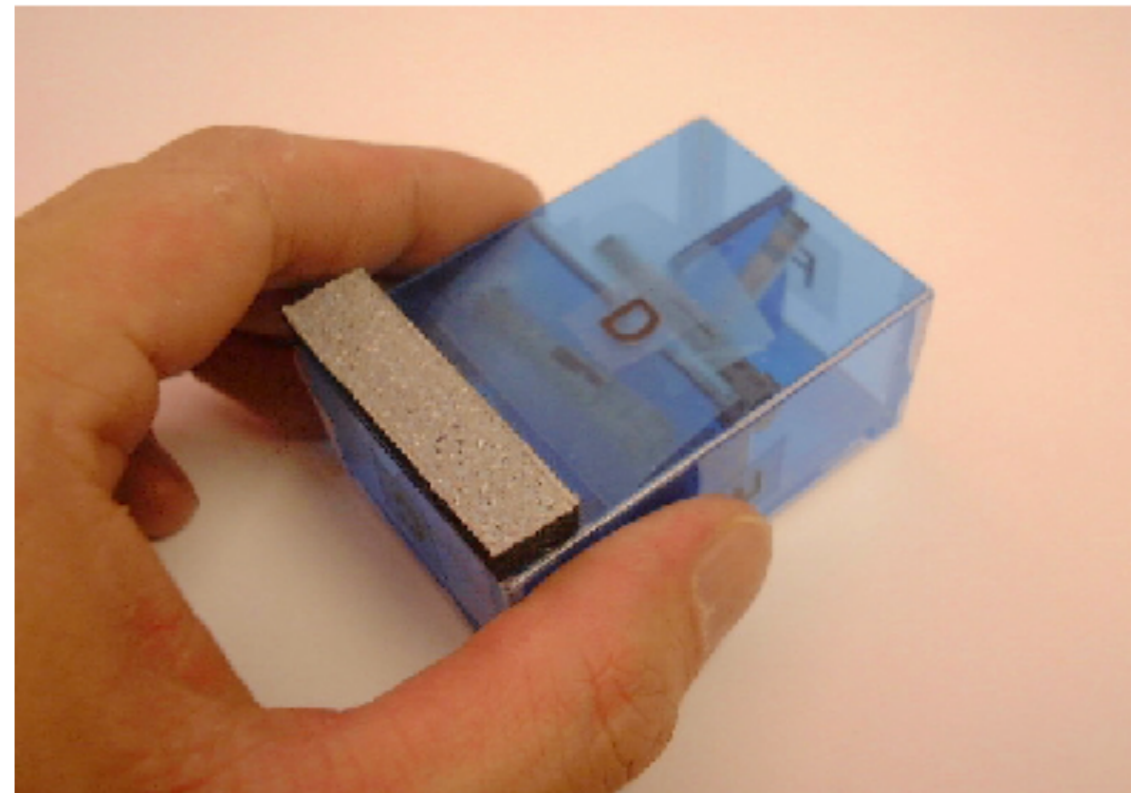




# ToolStone

(Rekimoto, Sciammarella, UIST 00)

- Universal 6 DOF input device
- Works on a Wacom pen tablet
- Can be used together with pens



# ToolStone working principle

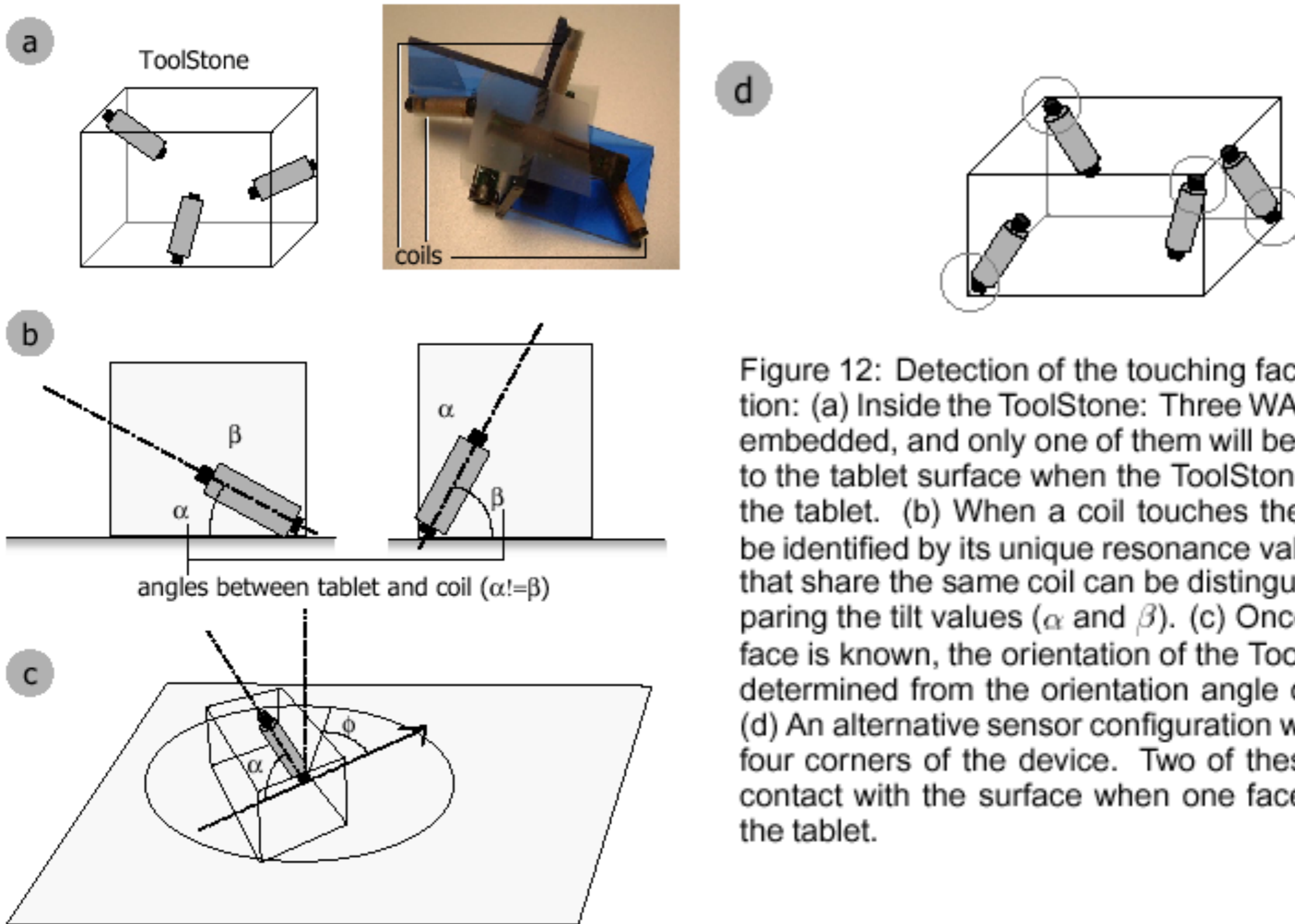


Figure 12: Detection of the touching face and orientation: (a) Inside the ToolStone: Three WACOM coils are embedded, and only one of them will be close enough to the tablet surface when the ToolStone is placed on the tablet. (b) When a coil touches the tablet, it can be identified by its unique resonance value. Two faces that share the same coil can be distinguished by comparing the tilt values ( $\alpha$  and  $\beta$ ). (c) Once the touching face is known, the orientation of the ToolStone can be determined from the orientation angle of the coil ( $\phi$ ). (d) An alternative sensor configuration with coils at the four corners of the device. Two of these coils are in contact with the surface when one face is placed on the tablet.

# ToolStone interaction

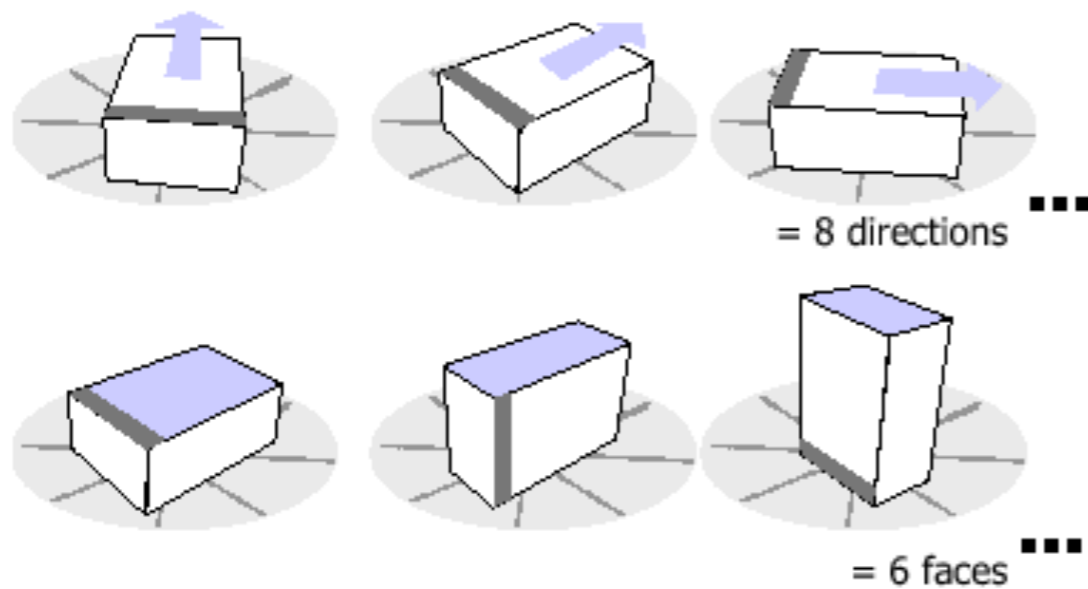


Figure 6: Selecting multiple functions by rotating and flipping the ToolStone: The combination of eight directions and six faces allows a user to quickly select 48 different functions (e.g., toolpalettes) with a single physical action.

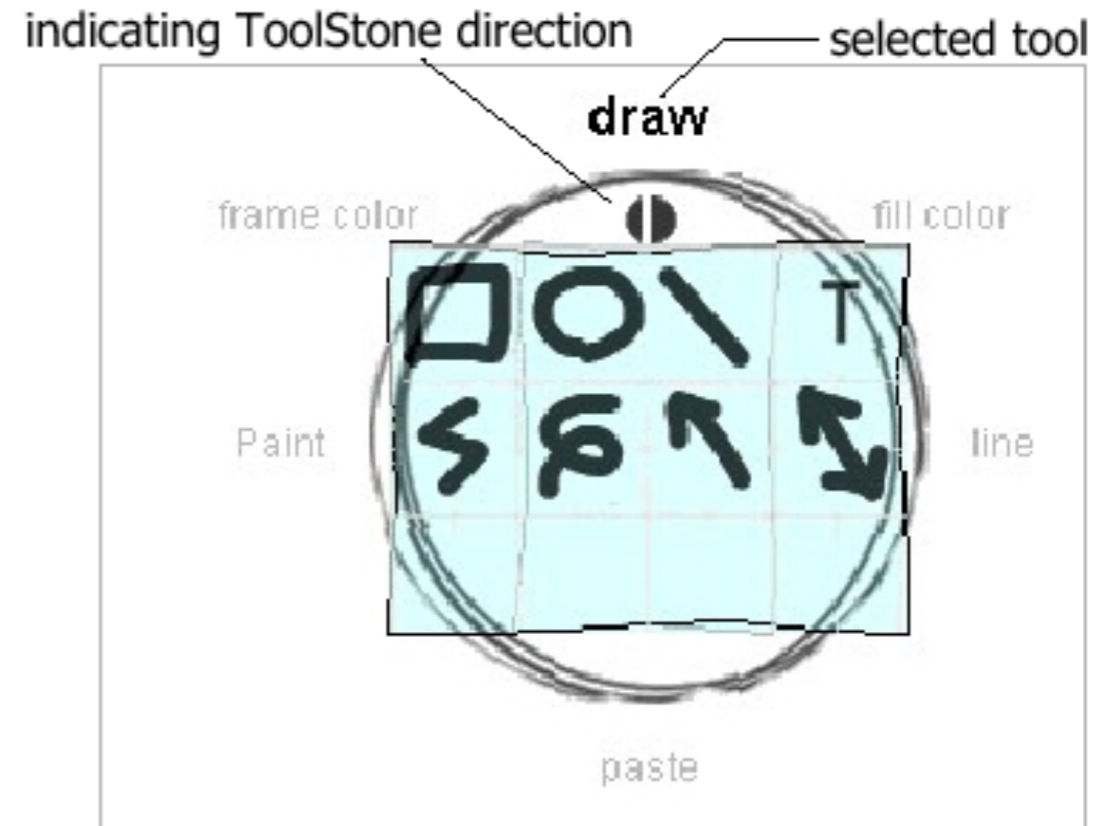


Figure 7: Example of a selected toolpalette: A dial and labels around the tool palette indicate available functionalities attached to the same face. The currently selected one is shown in bold. The selected toolpalette acts as a ToolGlass sheet.

# ToolStone interaction

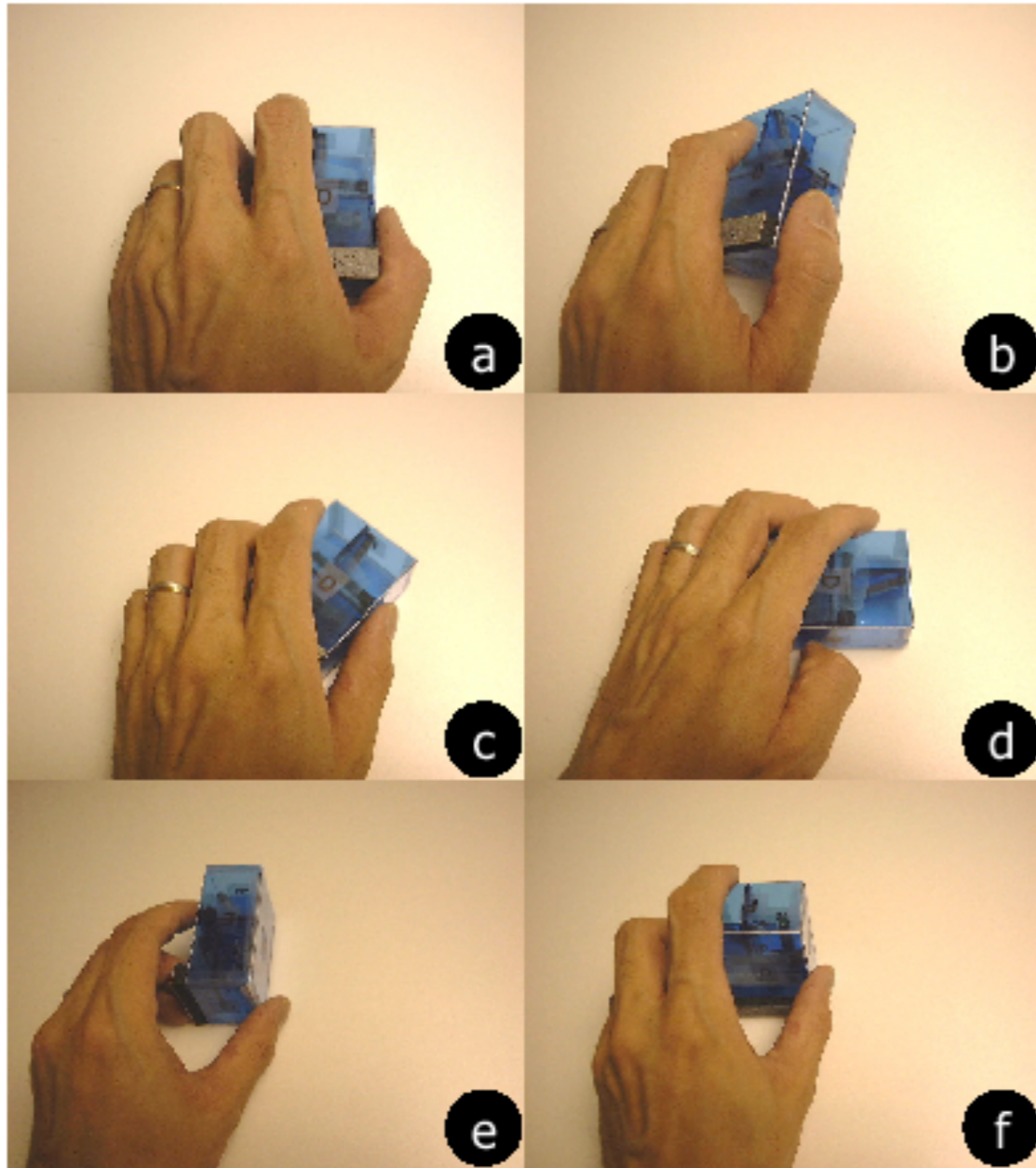


Figure 5: Several possible ways of holding the ToolStone: (a) Normal mode (Note: a projection attached near the lower edge of the upper face can be felt by the hand). (b) Tilting while one edge is contacting the tablet (c, d) Rotating, and (e, f) Flipping to select other faces.

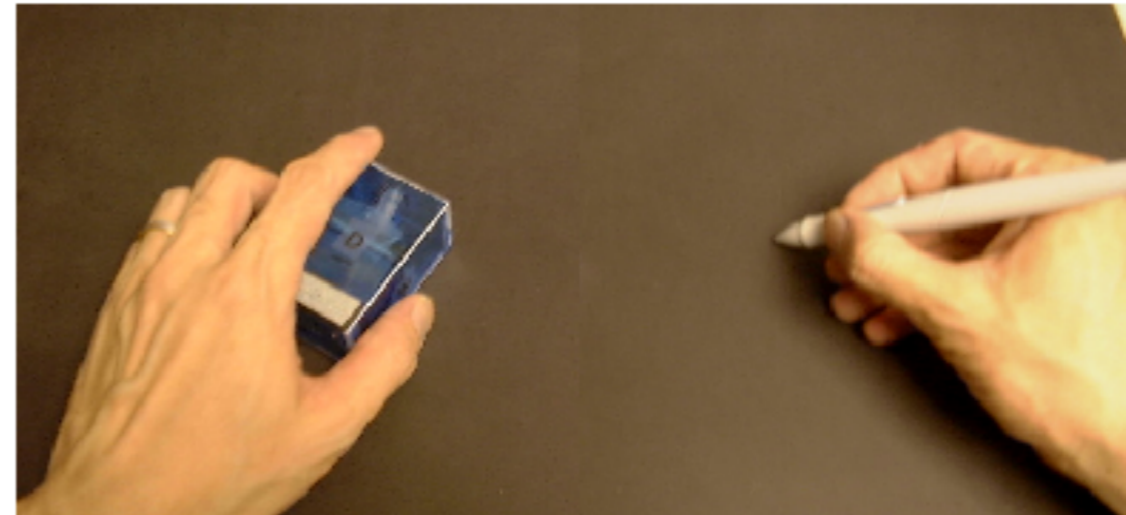


Figure 4: Bimanual interaction with the ToolStone.

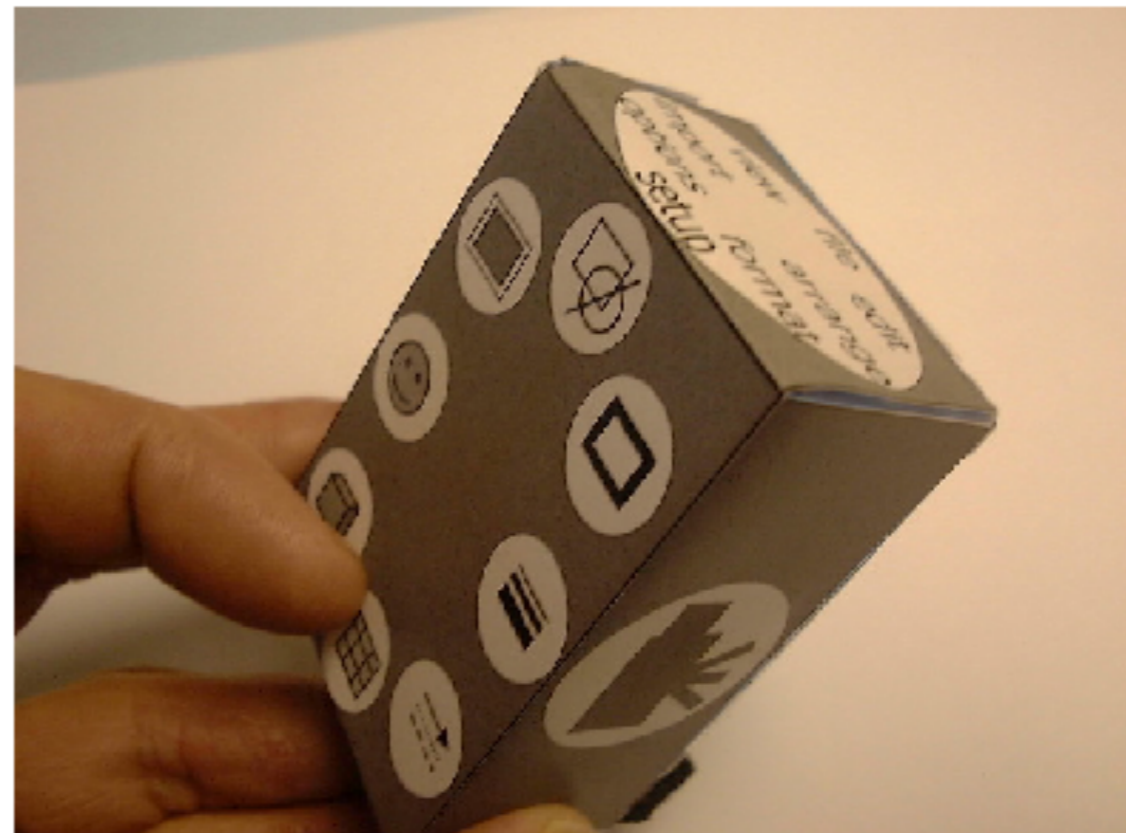


Figure 8: A ToolStone device with labels on each face. A (novice) user would be able to visually inspect available commands by physically turning the device.

# ToolStone interaction

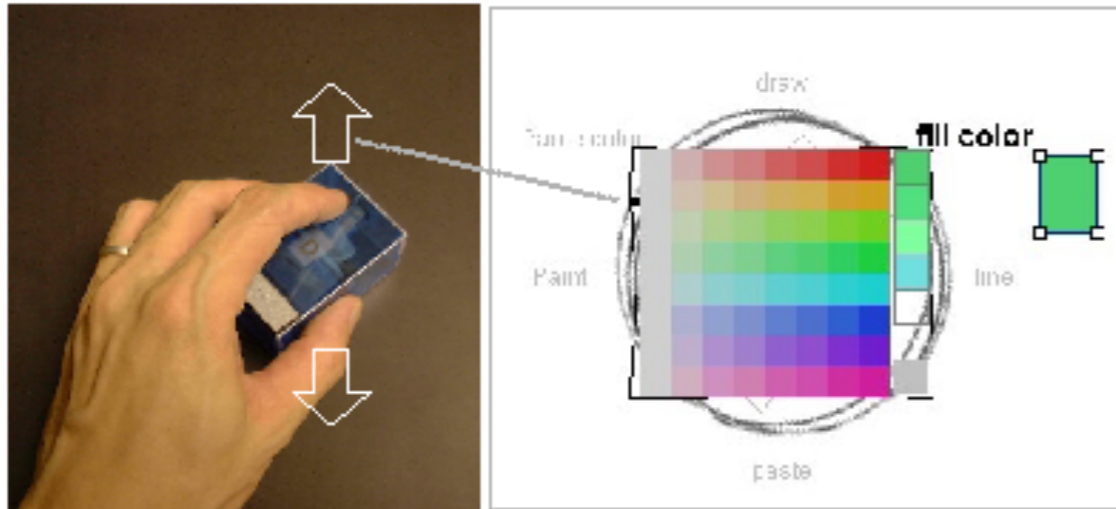


Figure 9: A color selection tool example: ToolStone's vertical motion controls the brightness parameter of the color space, while two other parameters (hue and saturation) are mapped according to the x and y axes of a 2D palette. A user can dynamically navigate through the color space before selecting a color instance. Note that the direction of the ToolStone is used to select the color selection tool.

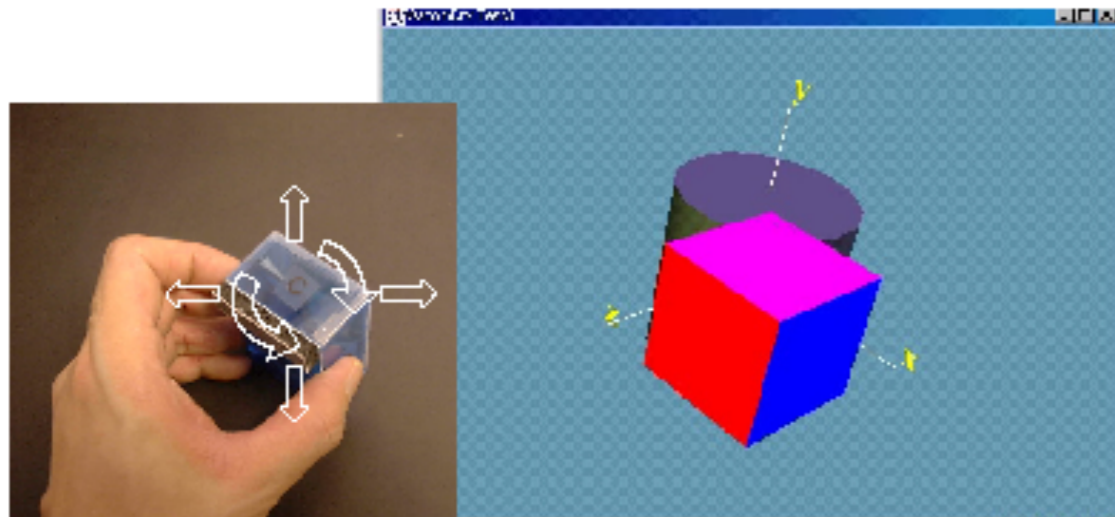


Figure 10: MDOF movement of the ToolStone can be mapped for 3D object control.

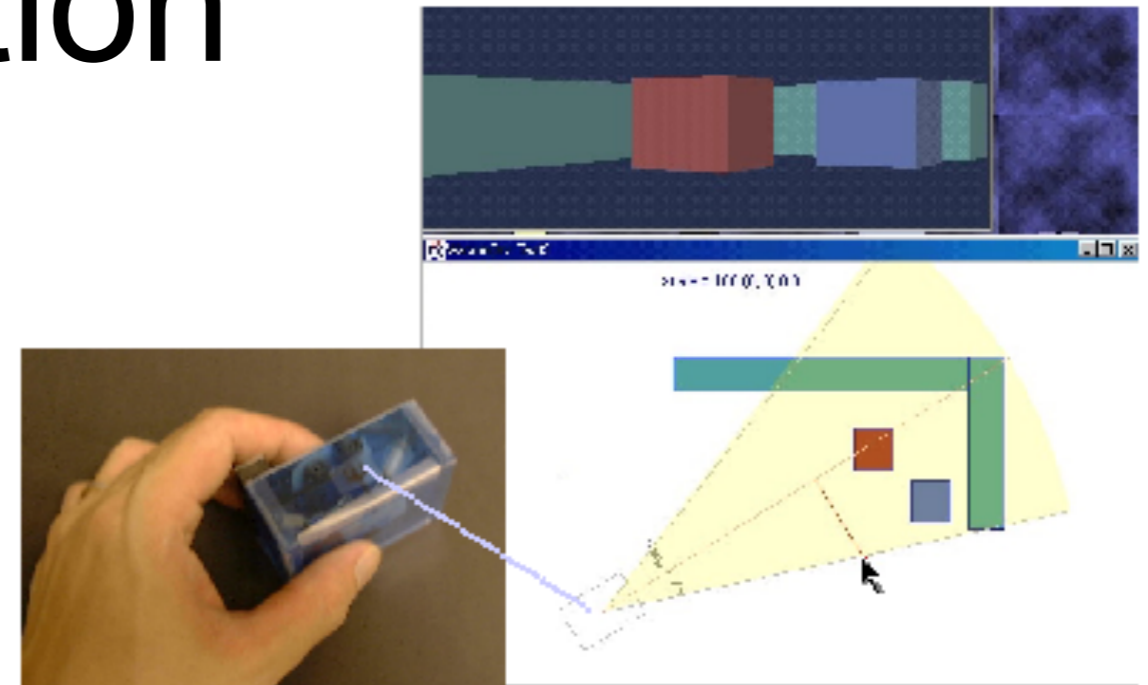
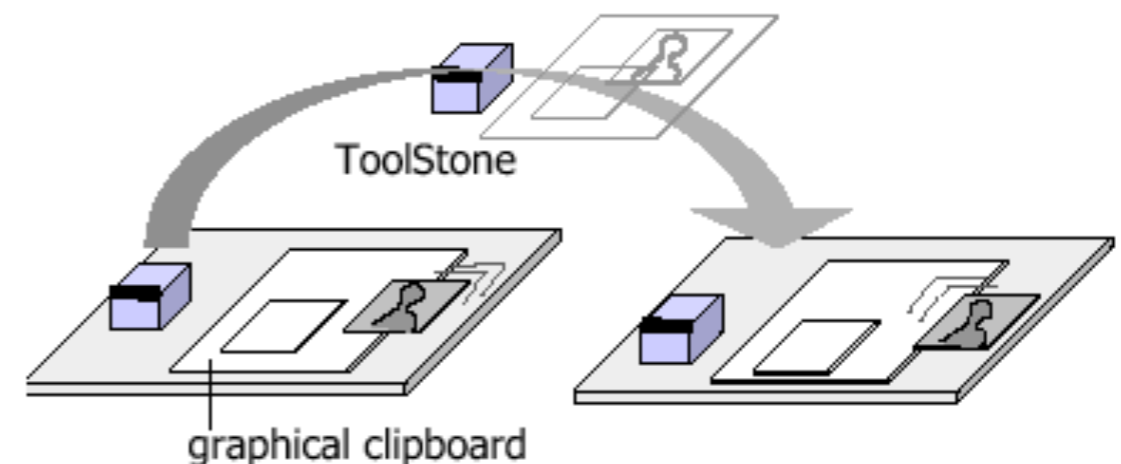
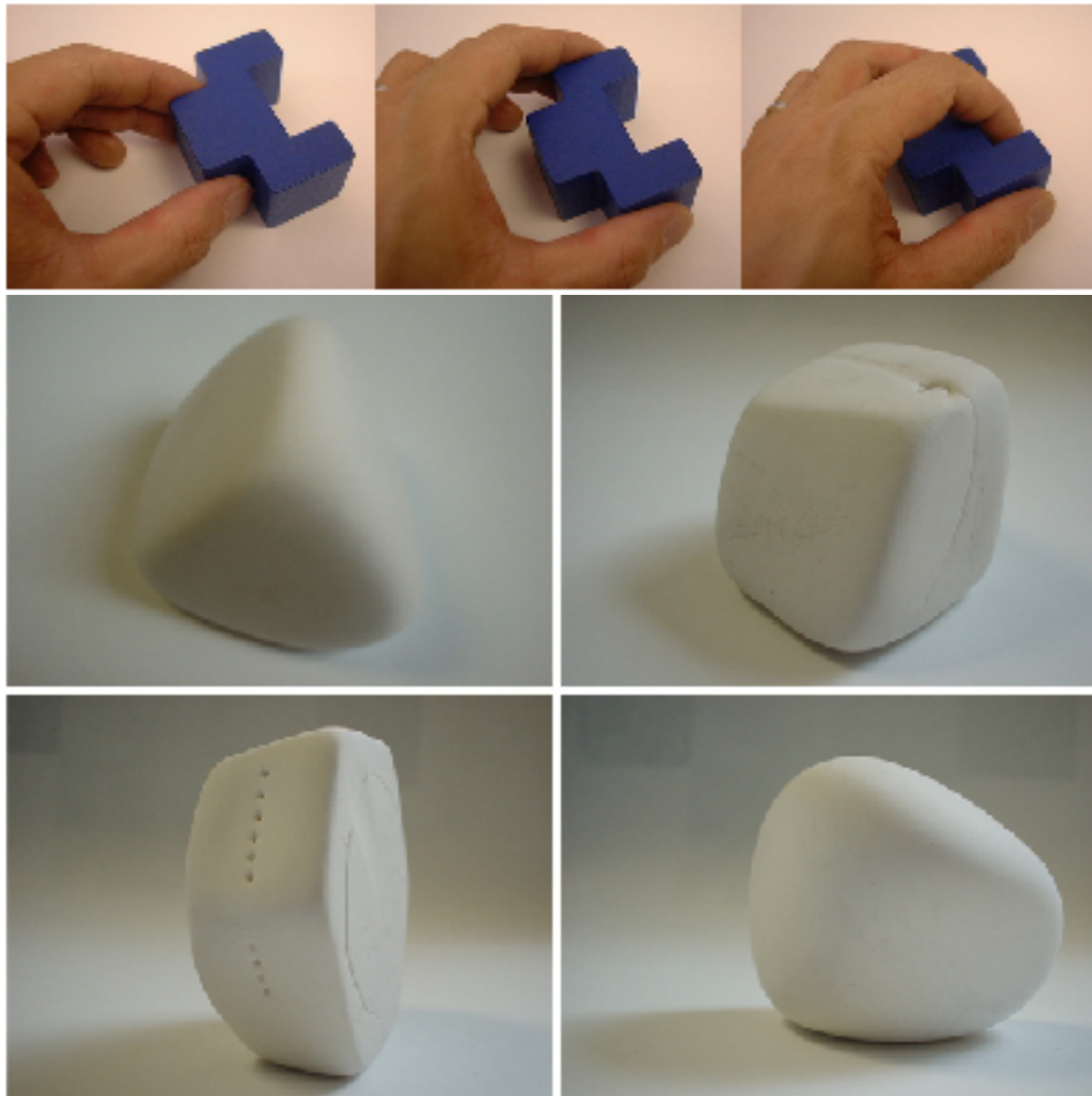


Figure 11: A user is manipulating a virtual camera of a 3D world. While the non-dominant hand is used to control the camera's position and orientation, the user can also change the field of view by dragging a viewing area (projected as a filled arc) with the dominant-hand's pointing device. Note that the pointing device is also used to change the viewing angle of the camera.



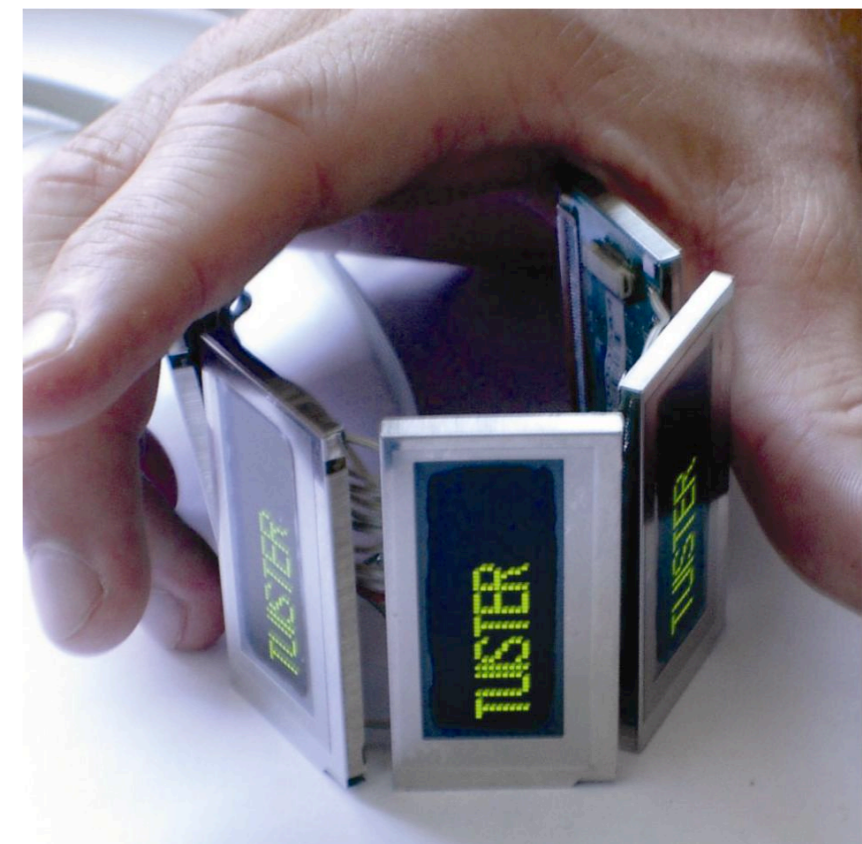
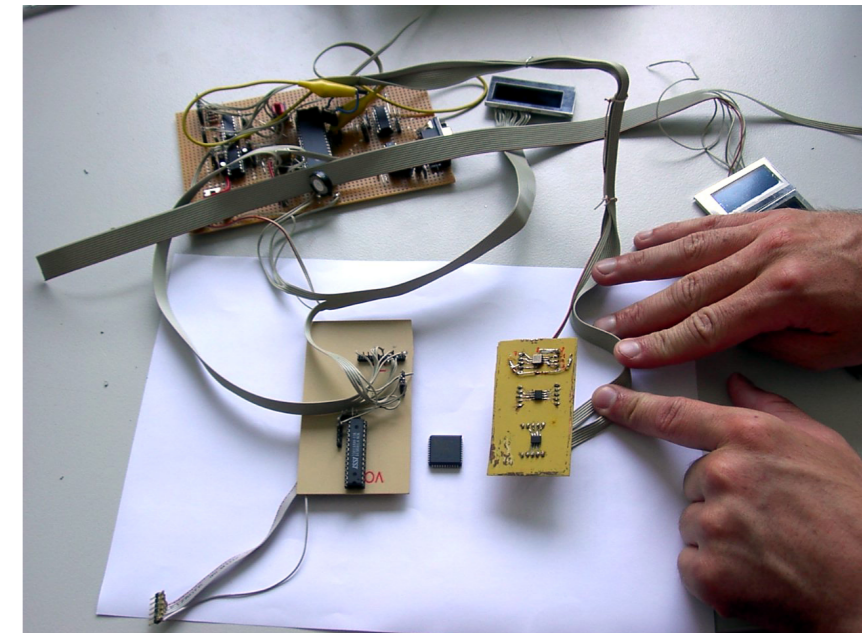
# ToolStone design variations



# TUISTER



- Interaction object, two-handed, 1DOF each
- Gravitation, magnetic and rotation sensors
- 6 organic Displays
- Serial/BT connection to the environment



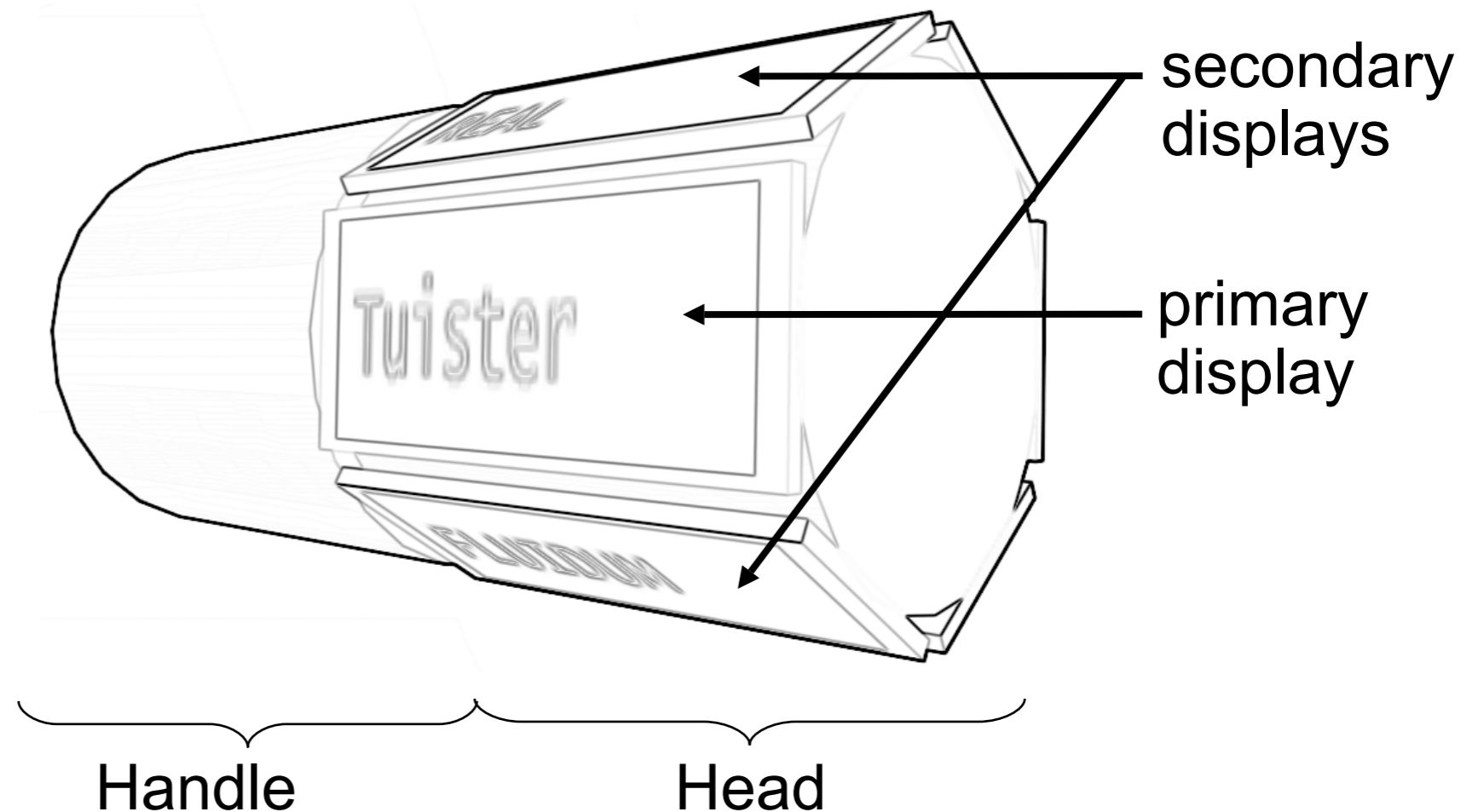
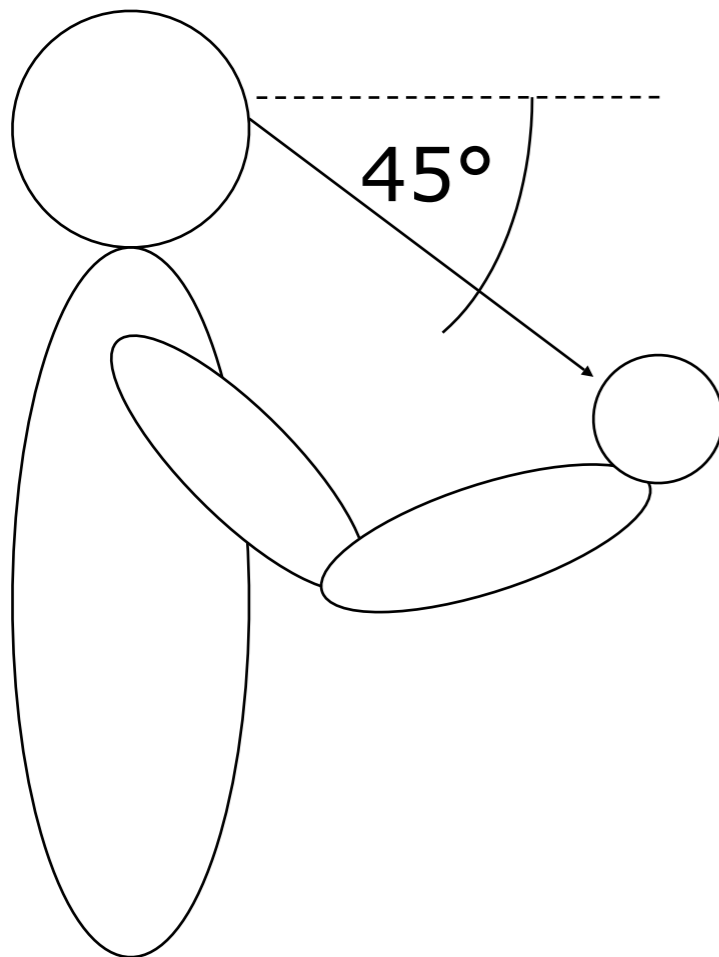
# Conceptual Design

Determination of the primary display by two assumptions:

- Text must be upright
- User looks down about  $45^\circ$

Sensors for orientation:

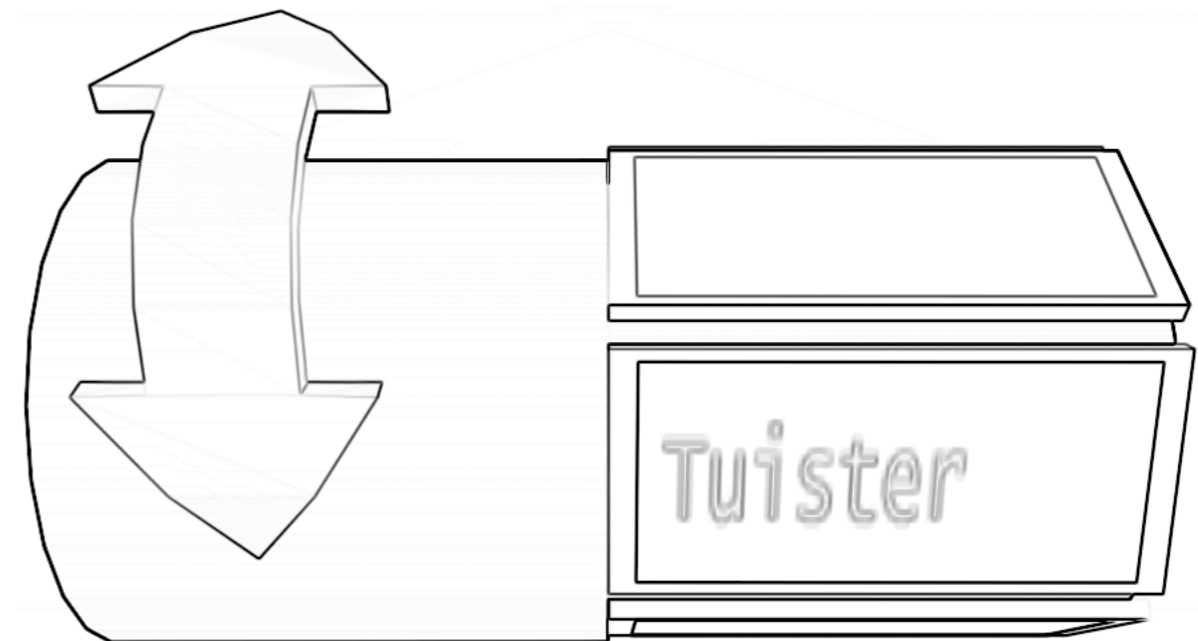
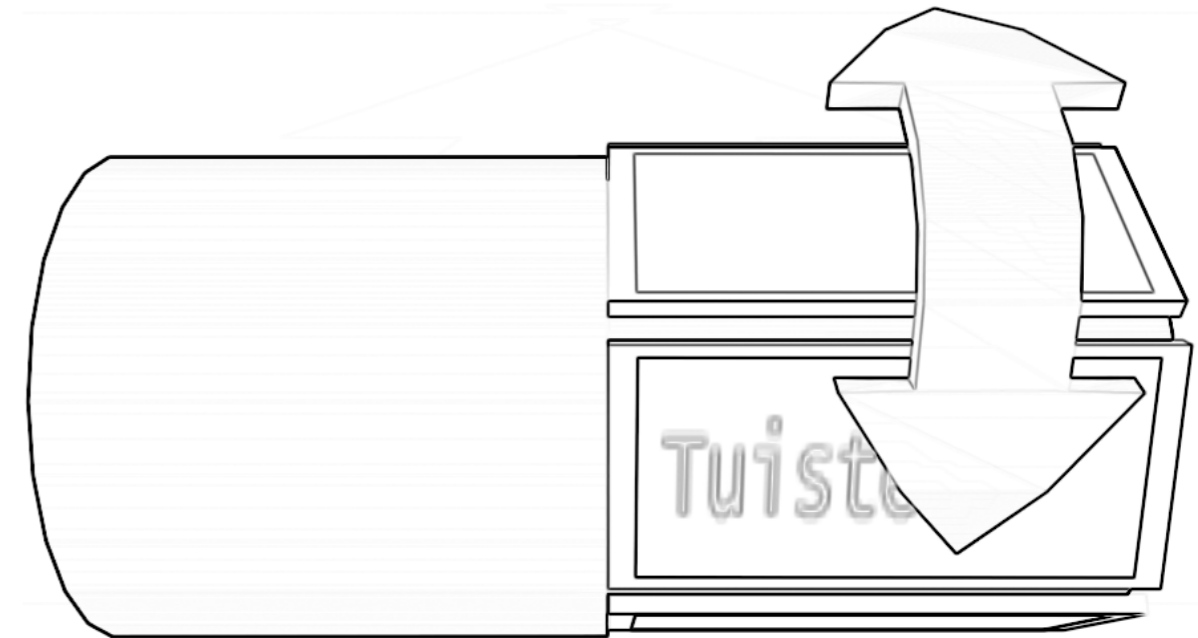
- 2x 2D acceleration
- 3x 1D magnetic
- 1x relative rotation





# Two types of rotation

- Rotating the head
  - Direct physical manipulation
  - Choice within one menu level
  - Context via secondary displays
- Rotating the handle
  - Metaphor: (un-)fastening a screw
  - Clockwise = fastening = down
  - Counterclockwise = up
  - Choice of the menu level
- Rotation by hand: few entries
- Free spin: for long menus

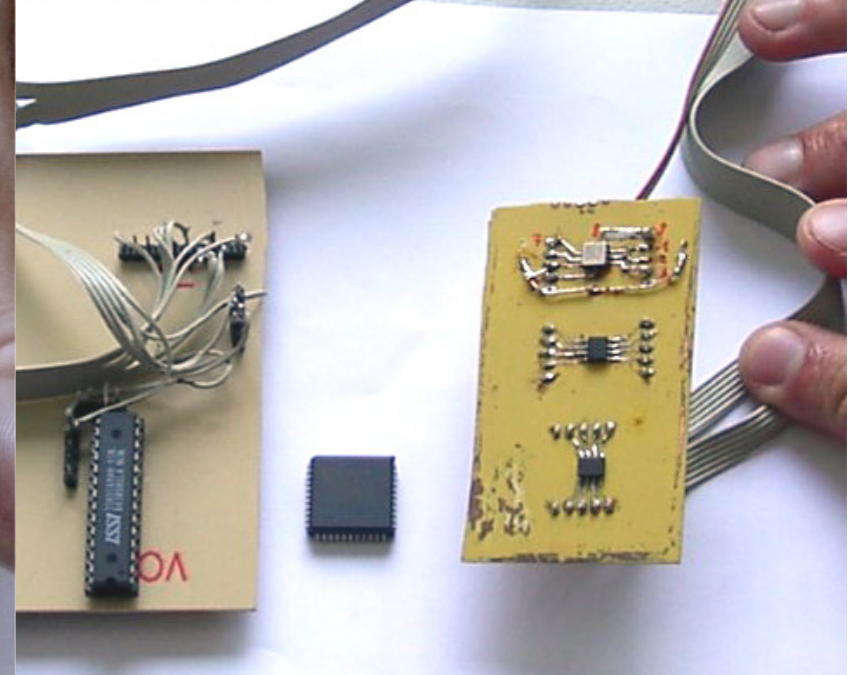
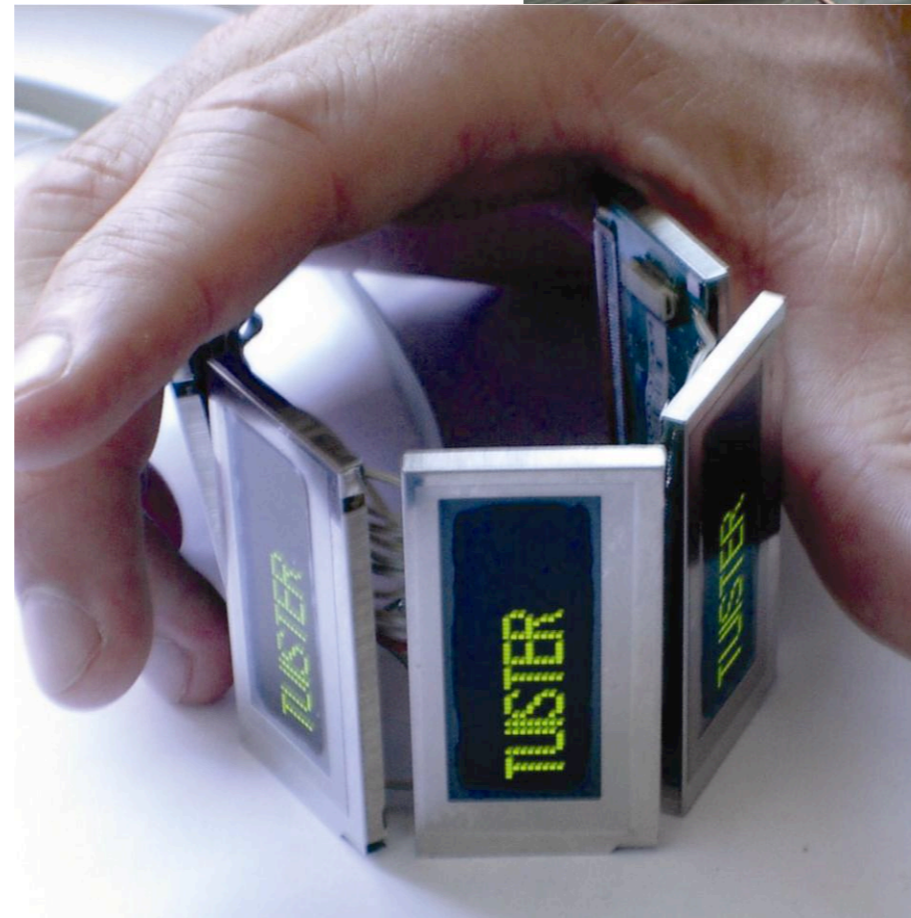
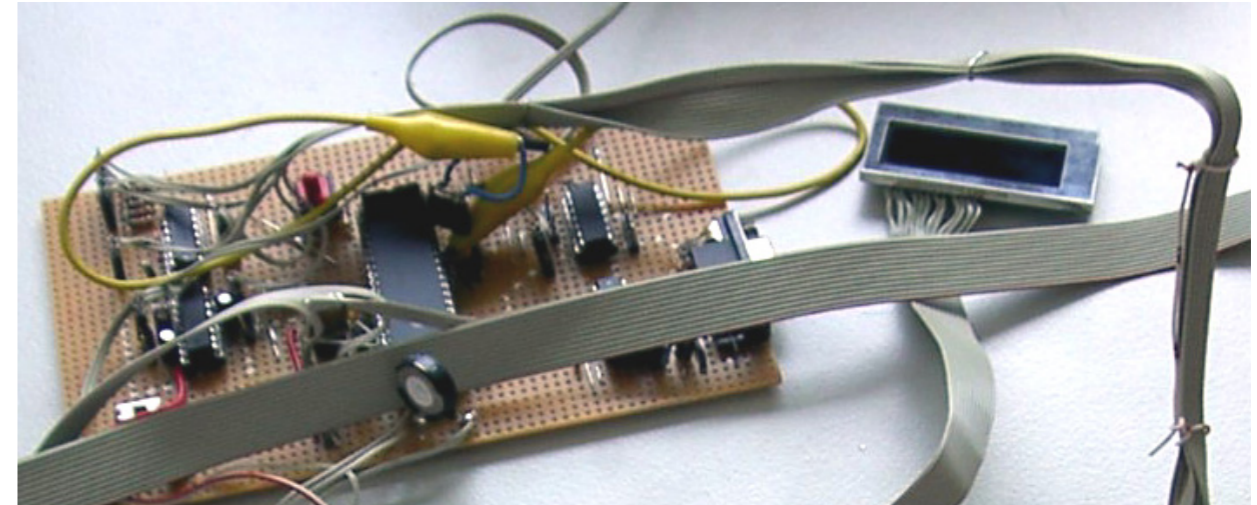
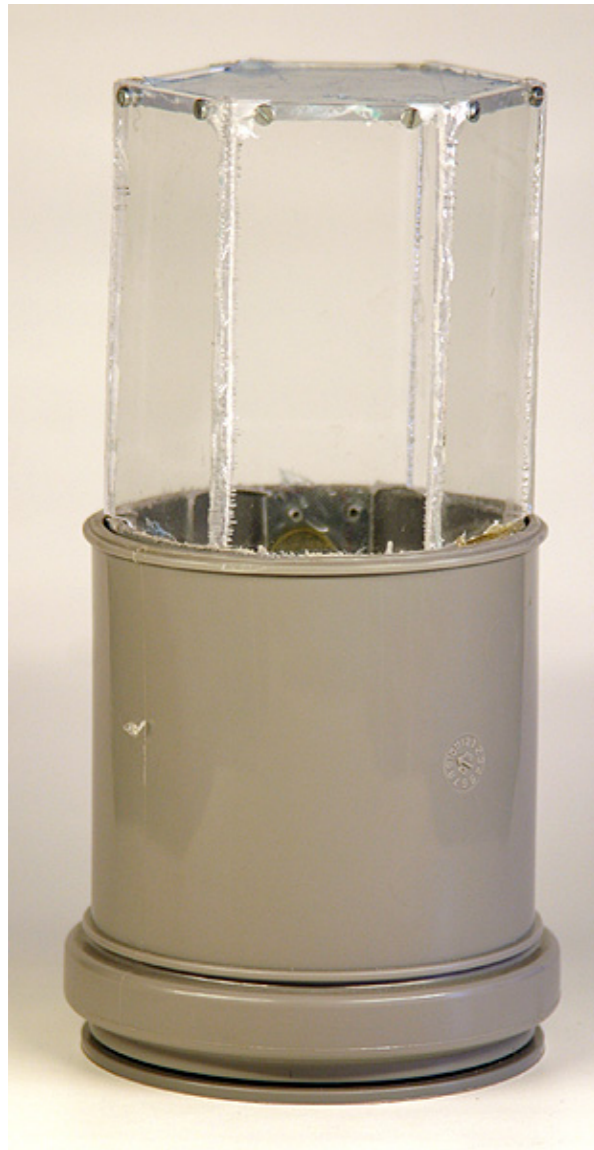


# Paper Prototype

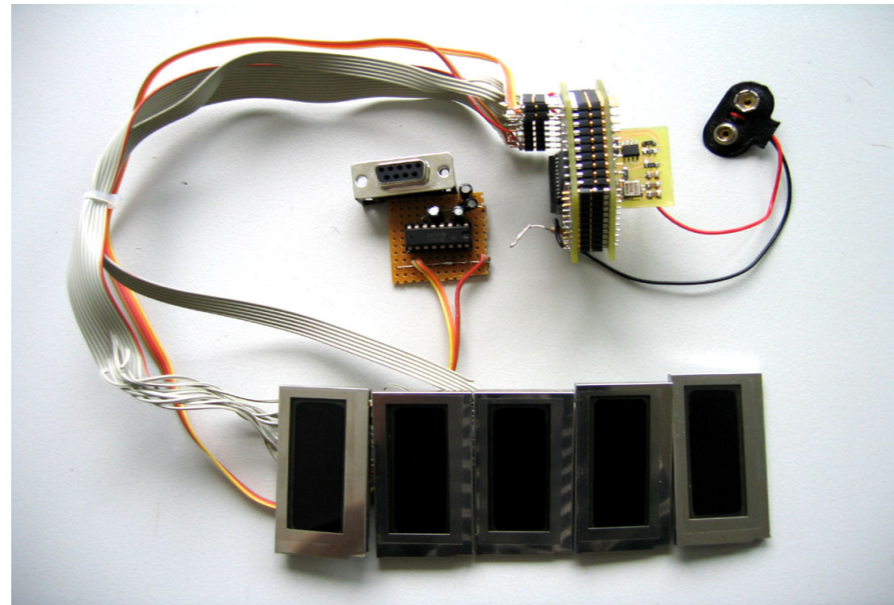
- Useful for discussing the concept
- No technical restrictions ;-)
- Close to the intended size



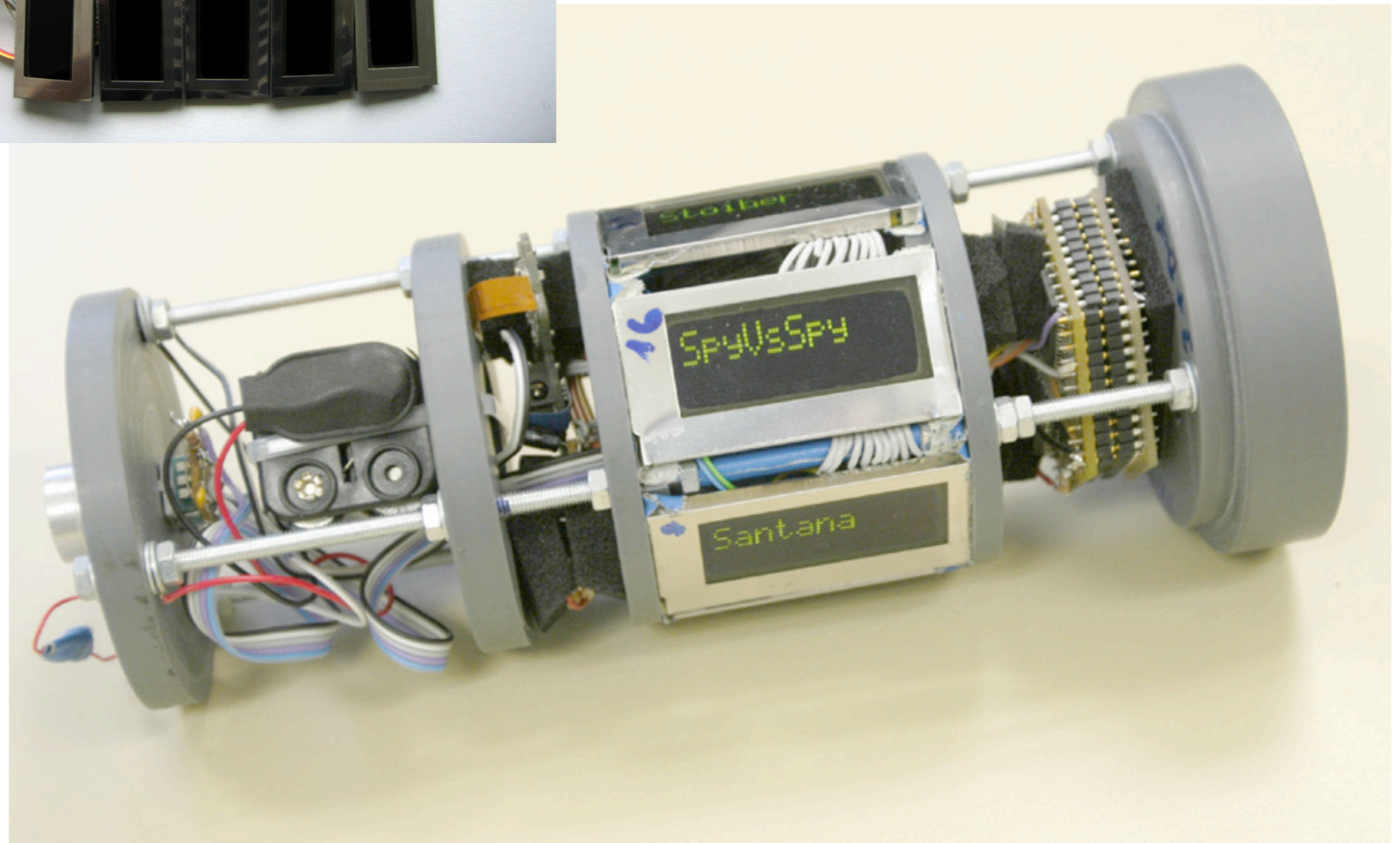
# Electronic Prototype



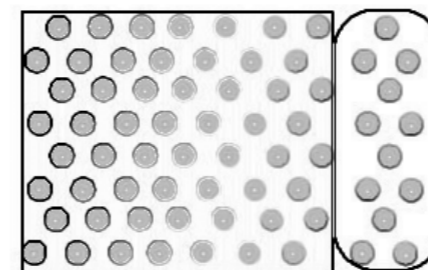
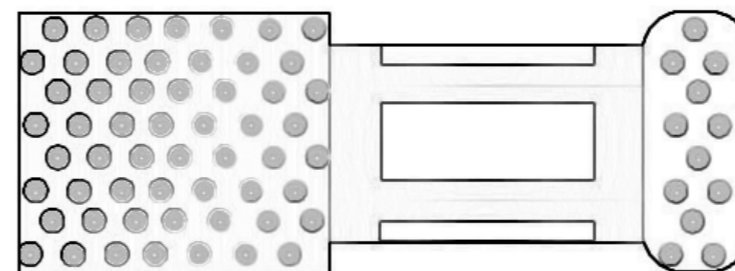
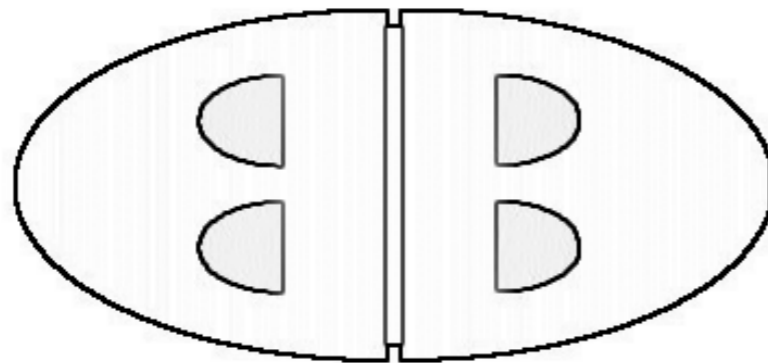
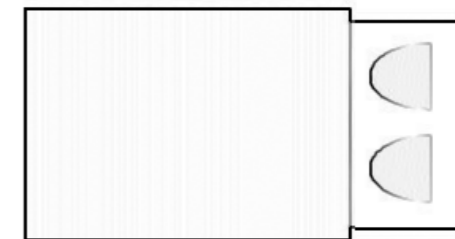
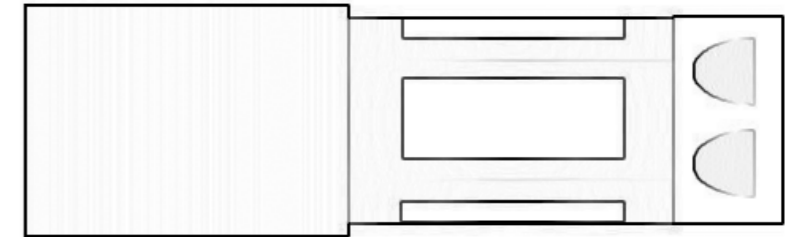
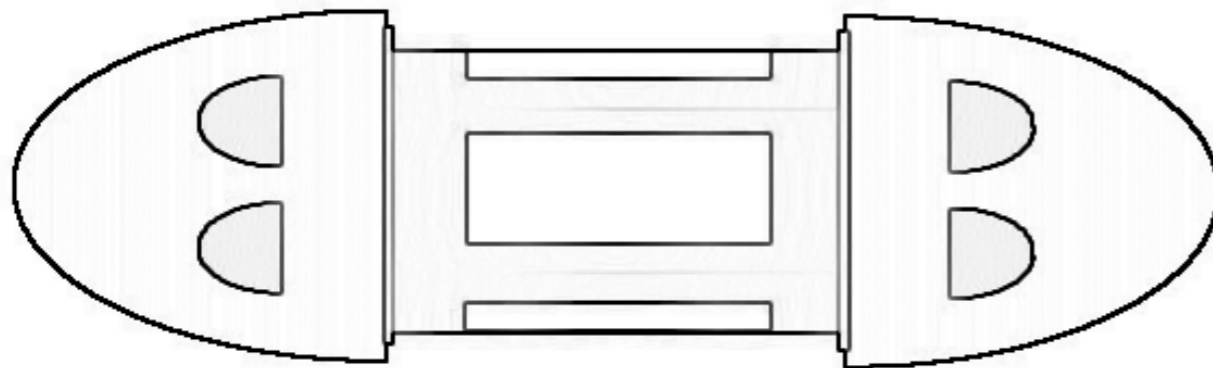
# First Functional Prototype



printed circuit boards  
bluetooth connection  
XML hierarchy descriptions  
mechanics very bulky

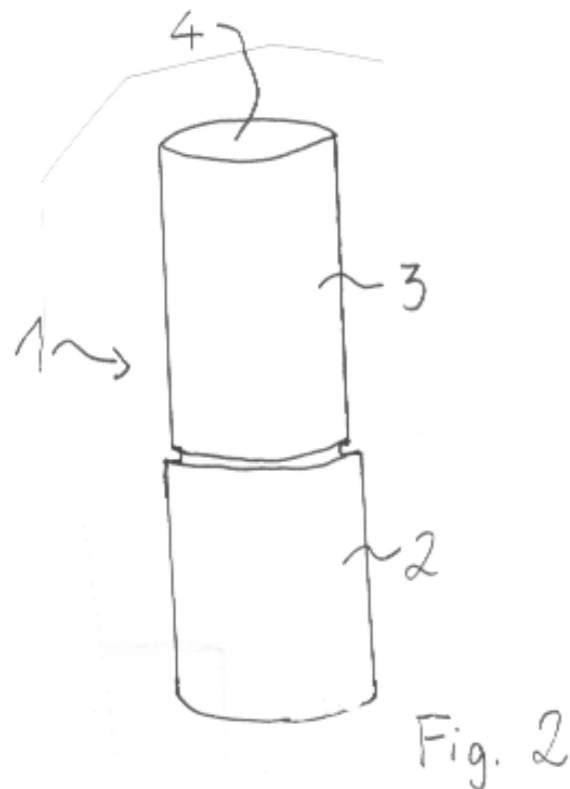
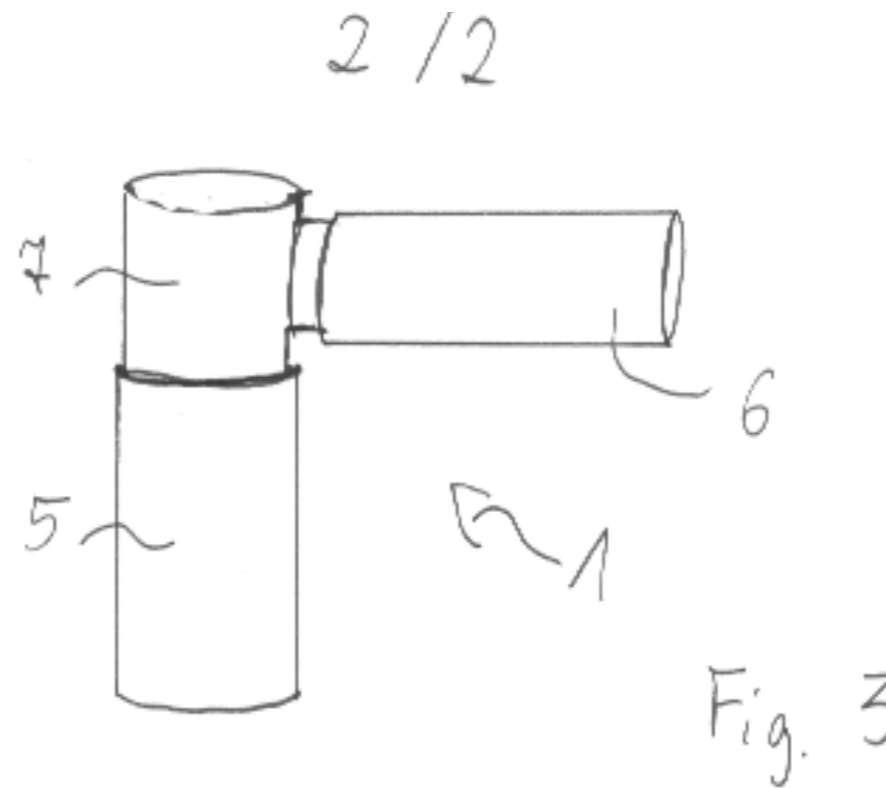
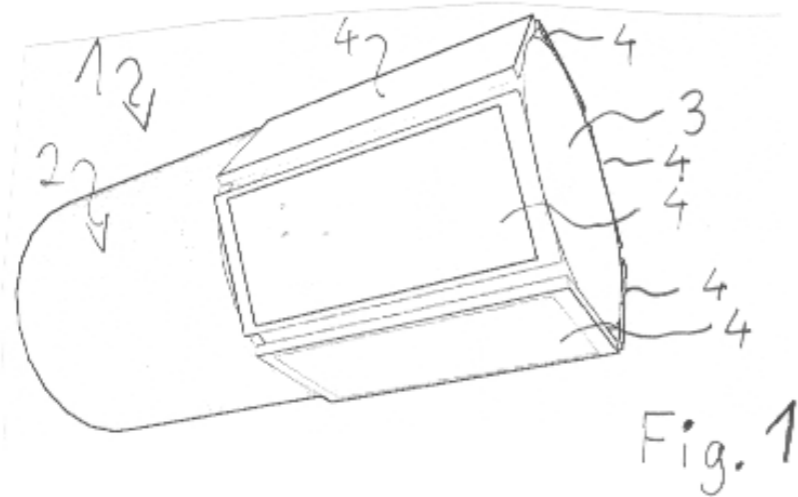


# Alternative Physical Designs



Courtesy of:  
Altmayer Design

# Generalization of the Concept



Abstraction of conceptual design  
Also includes display on front side  
Also includes non-coaxial designs

German Patent in 2003

# Tuister: Final prototype

