Using Visual Attention for Intelligent Multimodal UI

Ken Pfeuffer

About me

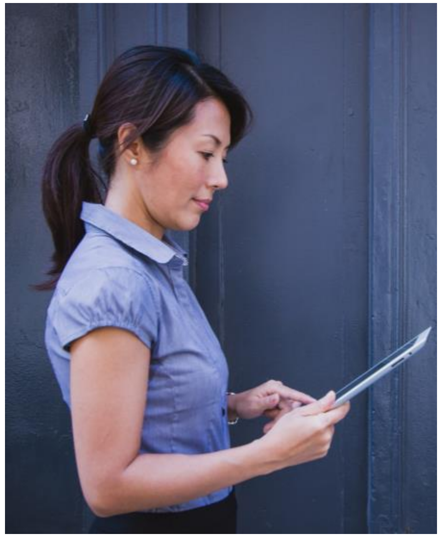
Half

Study

PhD

Internships

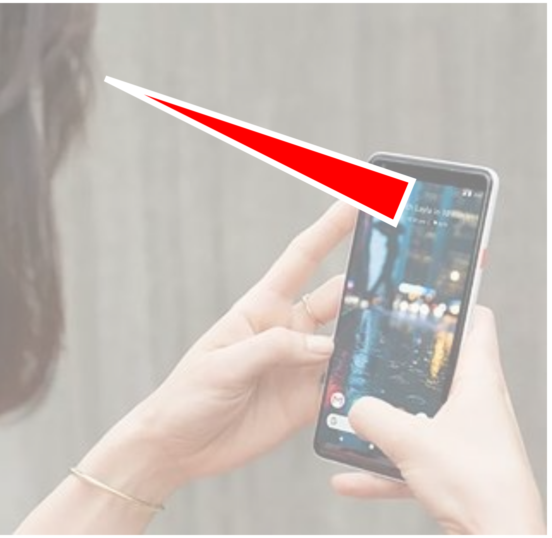

Phone


Tablet

Board


VR

Many UIs - One visual attention



Phone

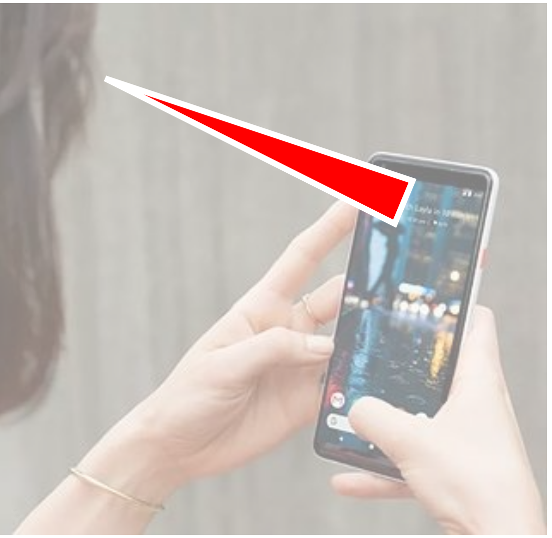
Tablet

Board

VR

To a human, the eyes are a perceptual channel, to get visual information.
To a computer, the eyes reveal visual and cognitive interest of the user.

Many UIs - One visual attention



Phone

Tablet

Board

VR

Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

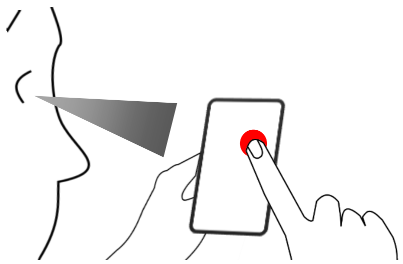
Implicit ⟷ Explicit

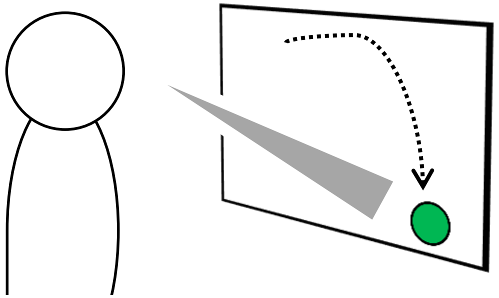
Many UIs - One visual attention

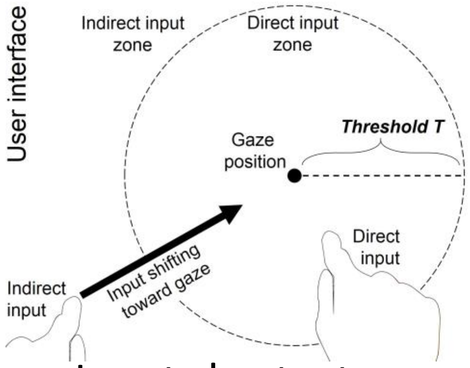
Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

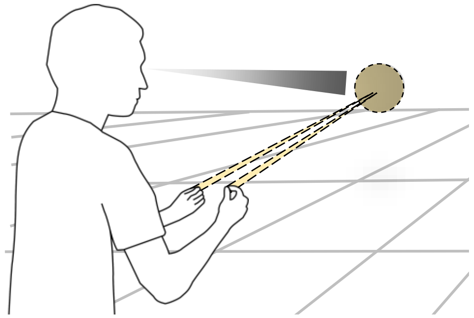
Implicit ⟷ Explicit


User performance modelling


Movement correlation & calibration



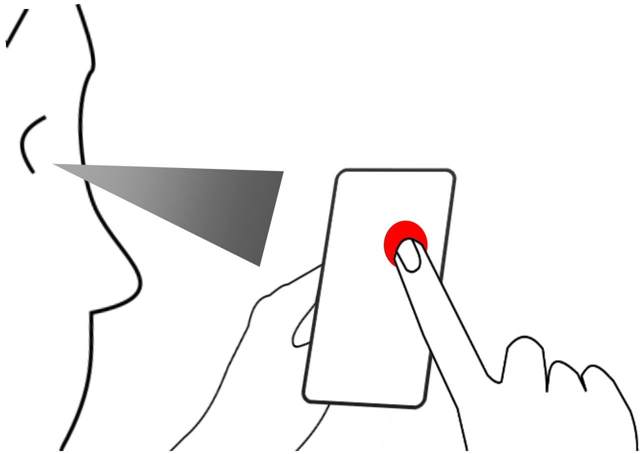
Input shortcuts


Gaze + Manual Input

Project 1: User Performance Modelling

Implicit

Data collection & offline analysis



Outline:

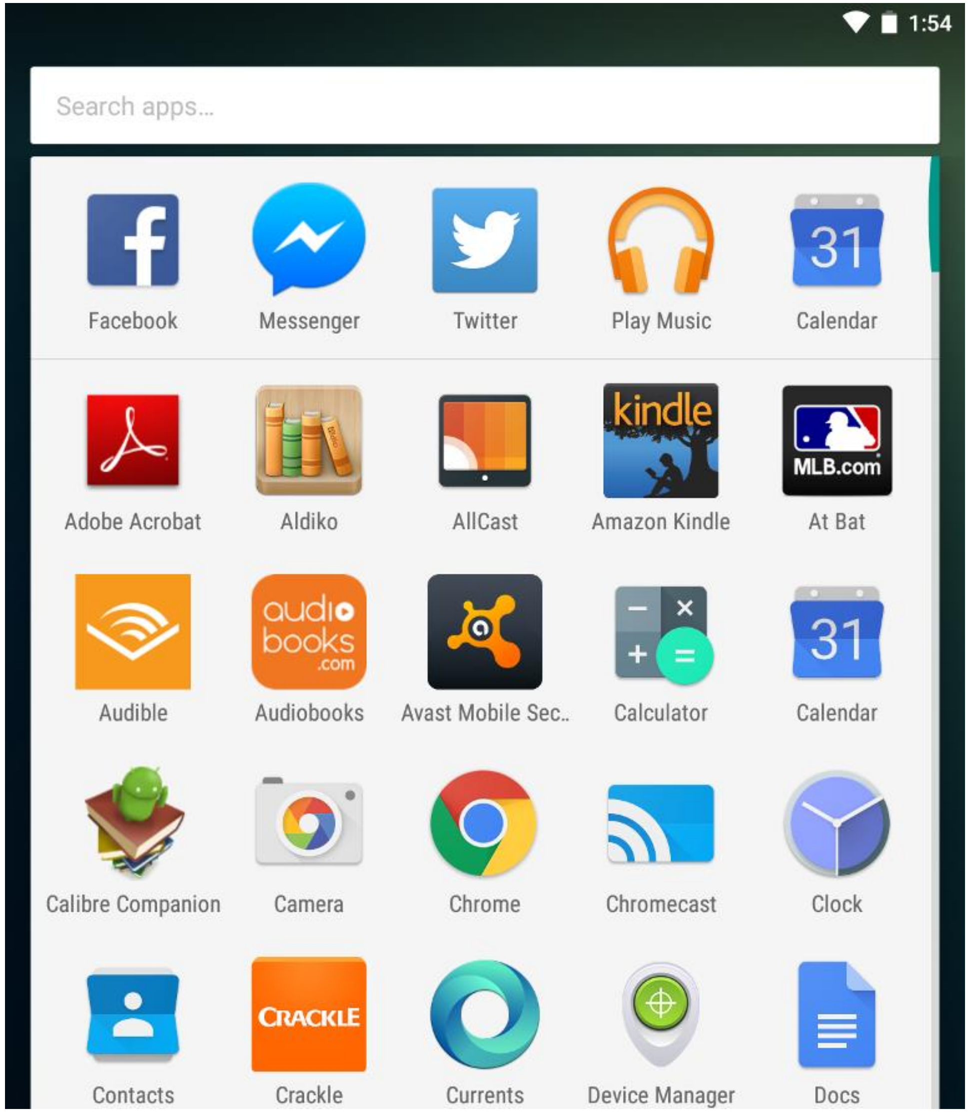
1. **Idea**
2. User study
3. Results
4. Model & Evaluation

The prediction bar

Users benefit by quick access of top5 predicted items.

Question 1: When do users benefit most?

Question 2: When do users benefit least?



⋮

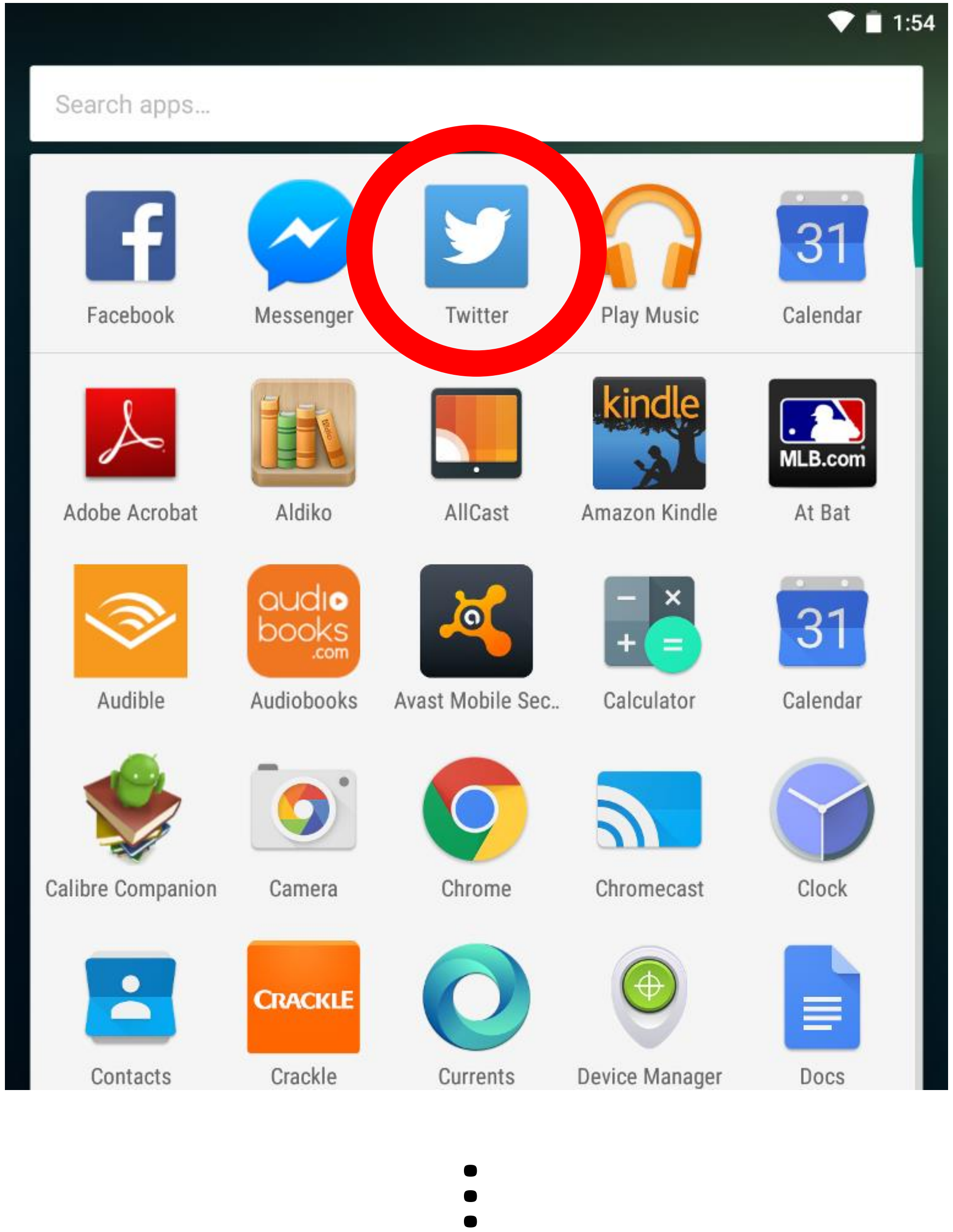
The prediction bar

Users benefit by quick access of top5 predicted items.

Question 1: When do users benefit most?

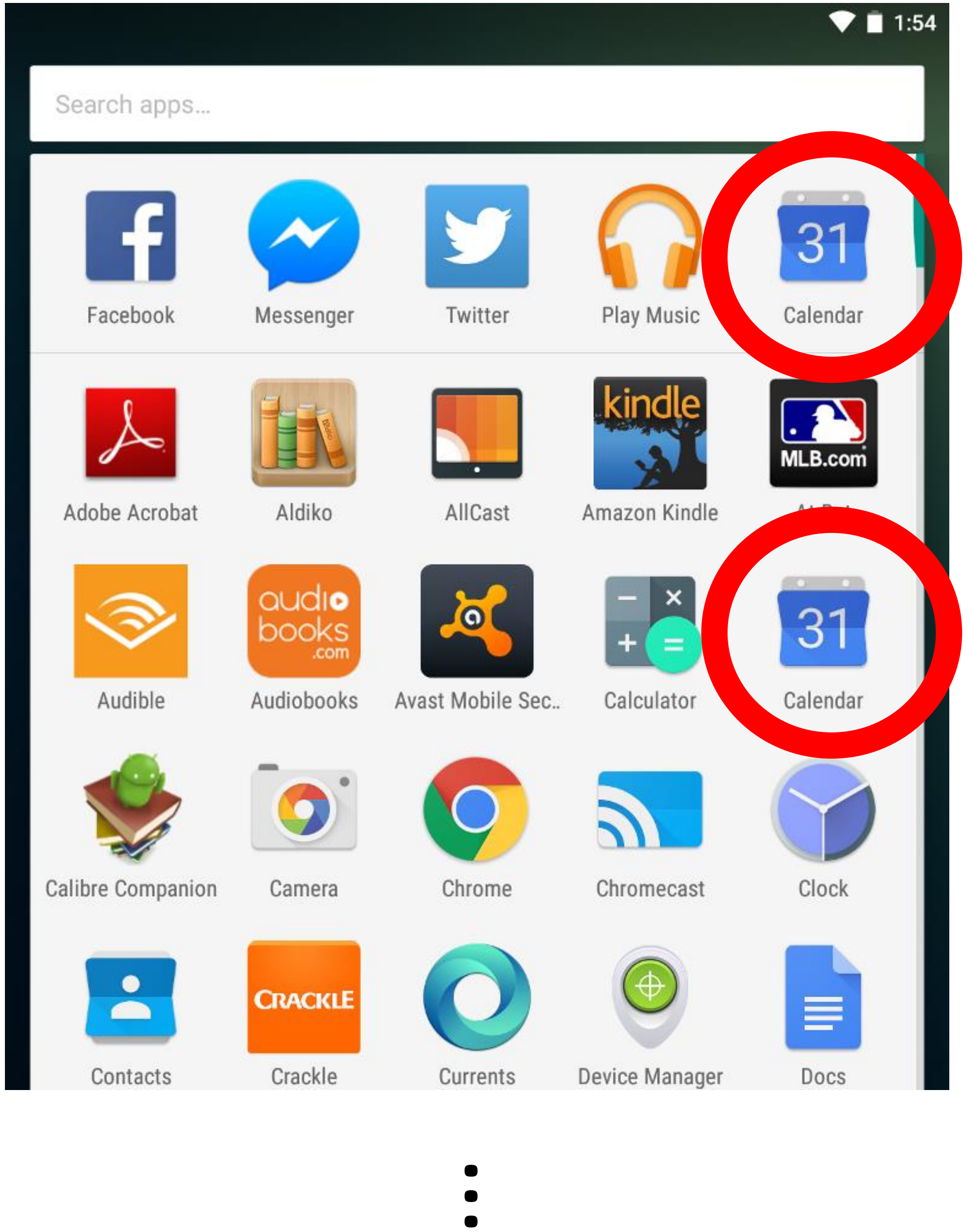
When the predicted items are far away.
Example: “Twitter”, where users scroll until “T”
→ High interaction cost

Question 2: When do users benefit least?



The prediction bar

Users benefit by quick access of top5 predicted items.



Question 1: When do users benefit most?

When the predicted items are far away.
Example: “Twitter”, where users scroll until “T”
→ High interaction cost

Question 2: When do users benefit least?

When the predicted items are very close.
Example: “Calender”, it’s on the same page!
→ Low interaction cost

Prediction benefit depends on interaction cost.

→ Incorporate interaction cost in prediction.

→ Use a model that predicts cost, i.e. app selection time.

→ What model?

Existing menu performance models

Pointing model Fitts' Law: pointing time depends on target distance & width.

- Only for last part of "touch" $T = a + b \log_2 \left(1 + \frac{D}{W}\right)$

Scrolling models: limited to mouse scrolling

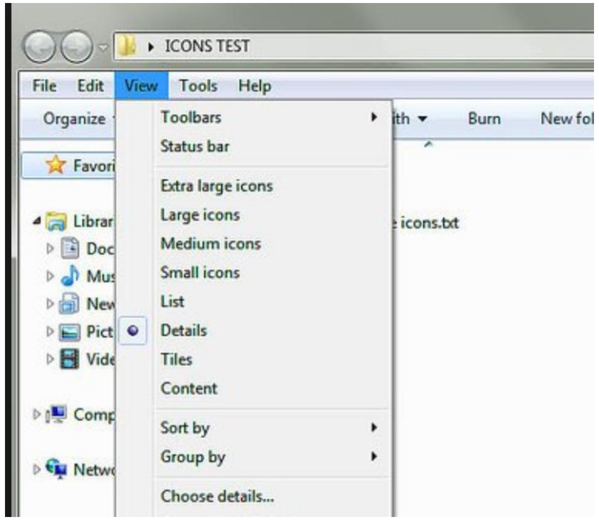
- Time increases linear with scrolling distance (when unordered)
- Time increases logarithmic with scrolling distance (when ordered)

Menu models: limited to linear desktop menus

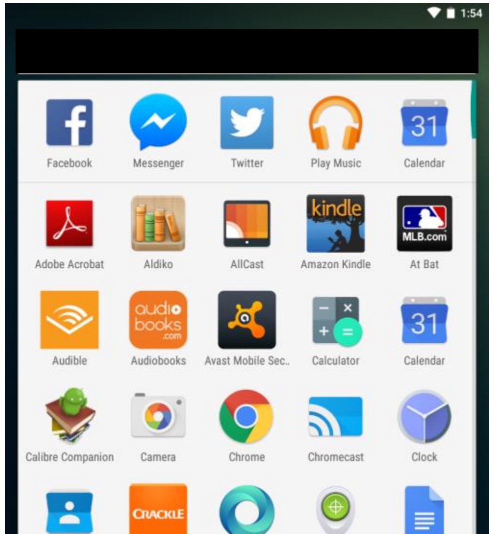
- Example SDP: **S**election, **D**ecision, **P**ointing --- Navigation?
- 2D grid menus?

Mobile != Desktop

1D, desktop



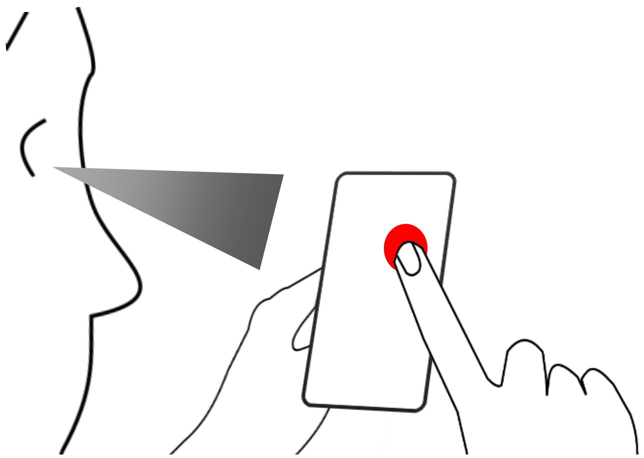
2D, mobile



Project 1: User Performance Modelling

Implicit

Data collection & offline analysis



Outline:

1. Idea
2. **User study**
3. Results
4. Model & Evaluation

User study

- 20 user
- Columns: 5 (fixed)
- Rows: 12, 18, 24, 30
- 8 blocks
- 15 trials per block

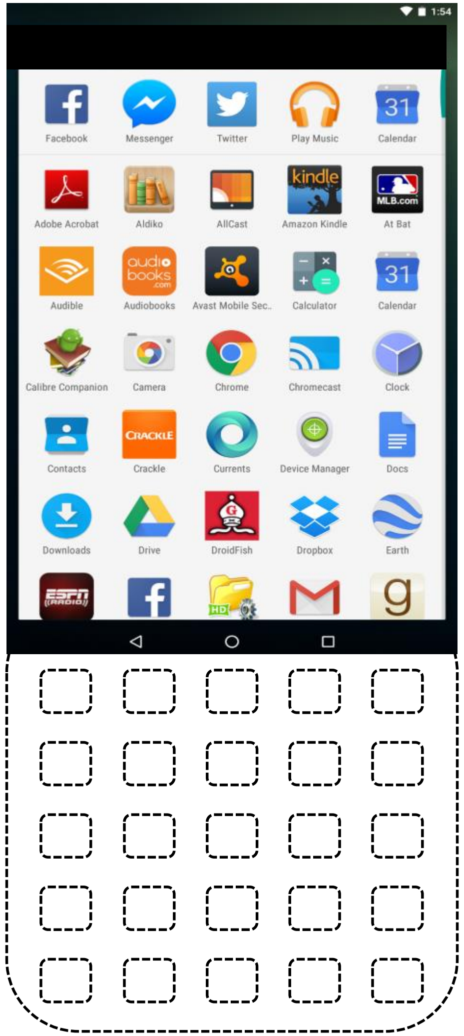
= 9600 trials

Nexus 6p, Tobii Glasses 2 eye tracker

- Rows: 12, 18, 24, 30

5 columns (fixed)

12 rows (variable)



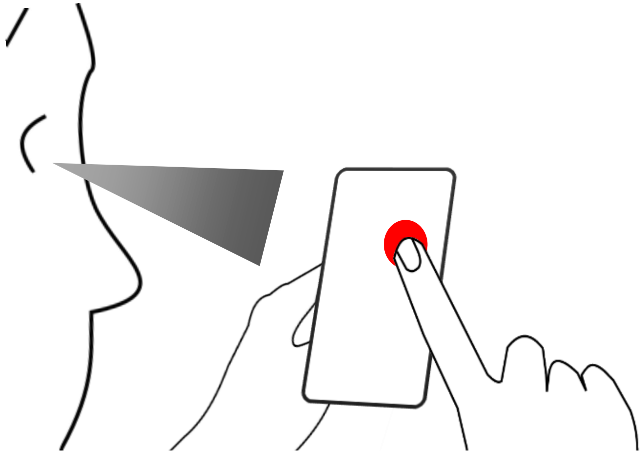
No scrolling needed

Scrolling needed

Project 1: User Performance Modelling

Implicit

Data collection & offline analysis



Outline:

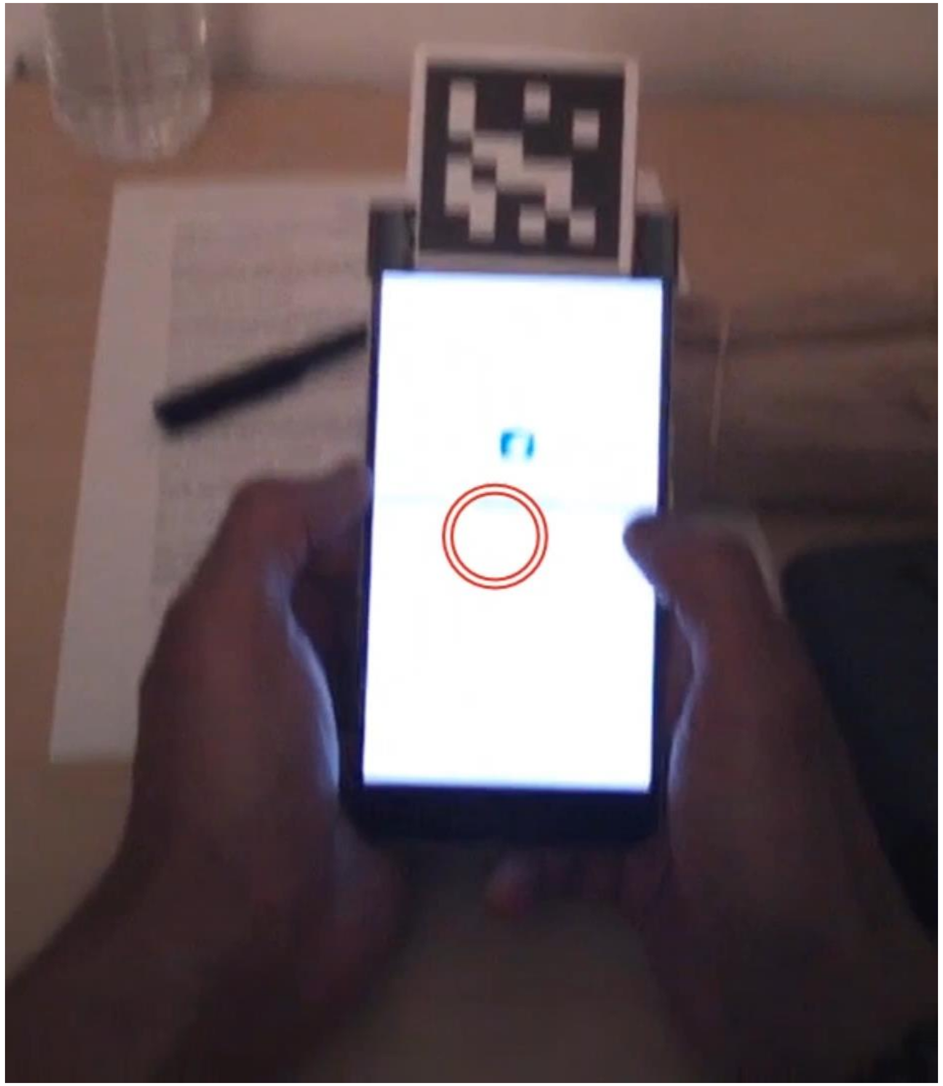
1. Idea
2. User study
3. **Results**
4. Model & Evaluation

Results

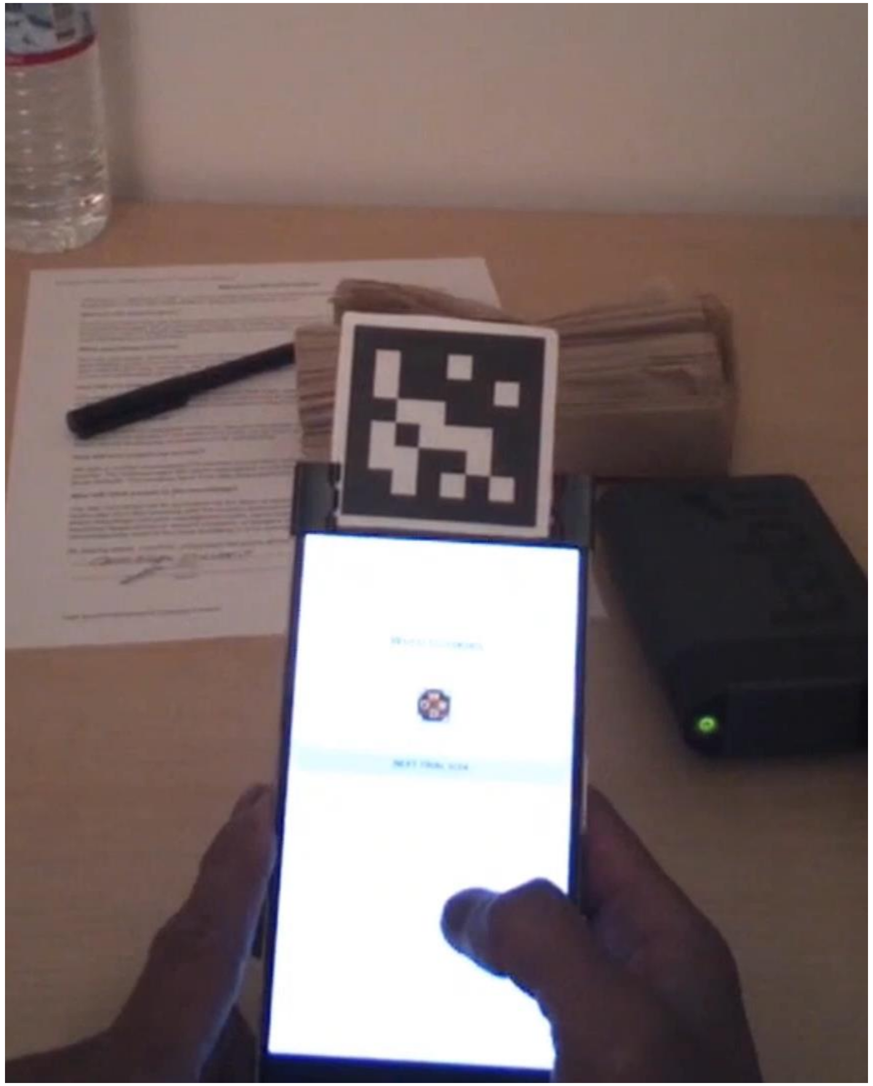
Results

Learning | Visual search | Navigation | Pointing

Block 1

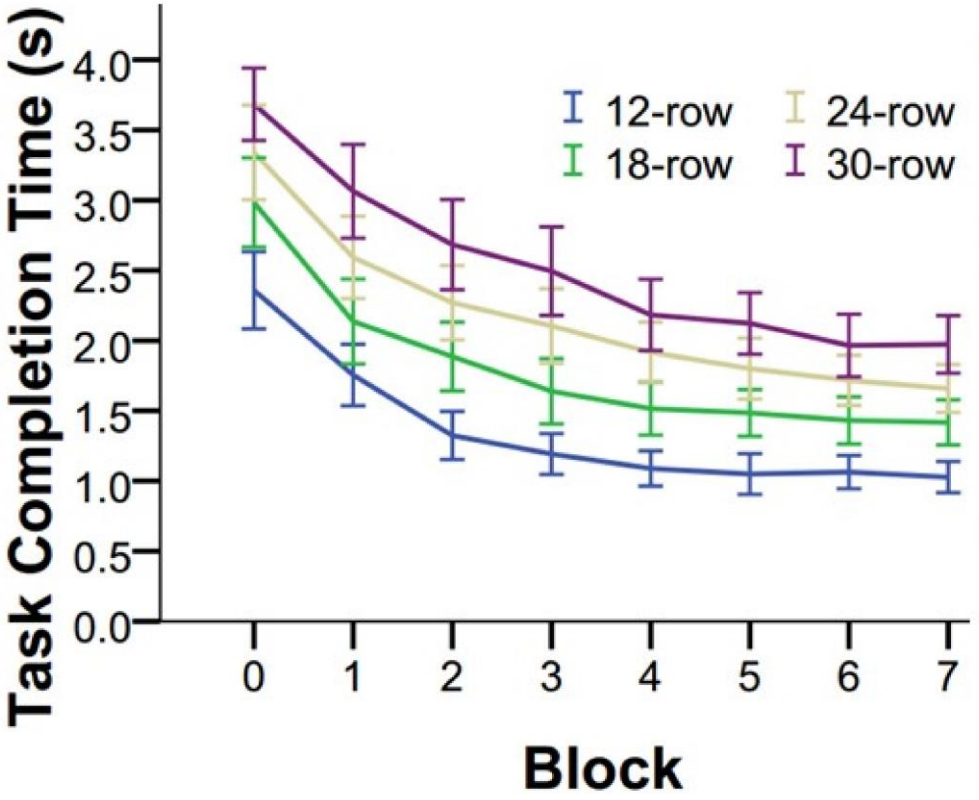


Block 8



Results

Learning | Visual search | Navigation | Pointing



To take into account the effect that users become better at examining each row with practice, the time incorporates the learning rate that decreases logarithmically with experience, modeled by the power law of practice*:

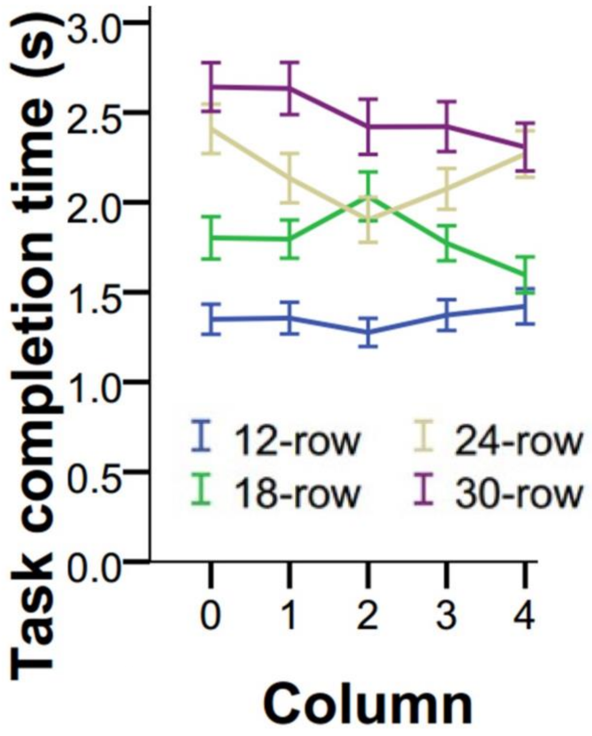
$$T_{row} = a_r \times e^{(-b_r \times t)} + c_r$$

where t denotes the number of previous trials, and a_r, b_r, and c_r are parameters to be learned.

* Based on formula in:
Gilles Bailly, Antti Oulasvirta, Duncan P. Brumby, and Andrew Howes. 2014. Model of visual search and selection time in linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 3865-3874. DOI: https://doi.org/10.1145/2556288.2557093

Results

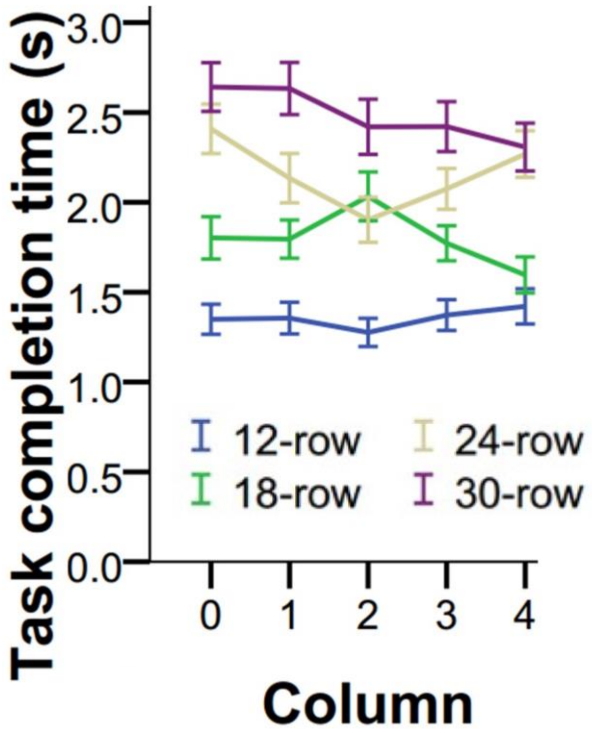
Learning | Visual search | Navigation | Pointing



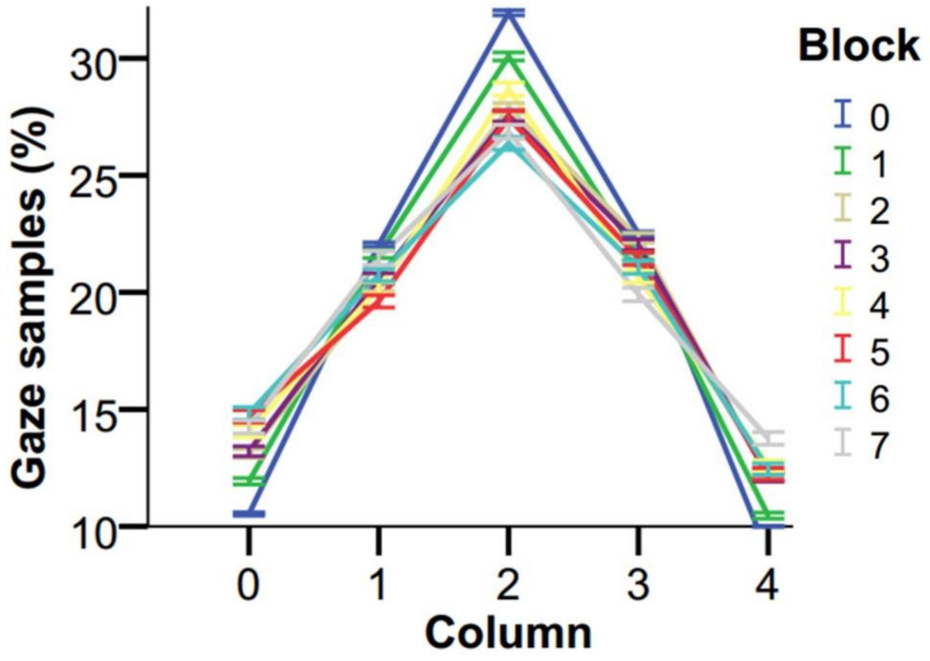
→ No statistical differences, but some tendency to center.

Results

Learning | Visual search | Navigation | Pointing

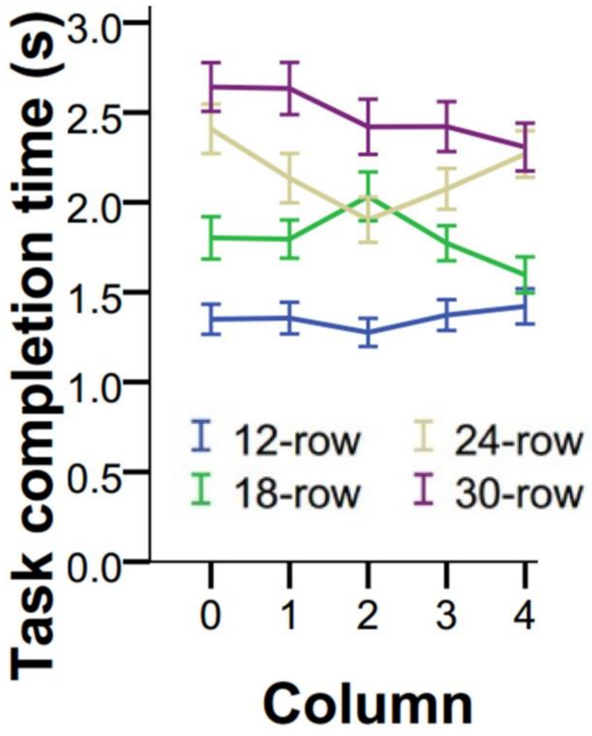


→ No statistical differences, but some tendency to center.

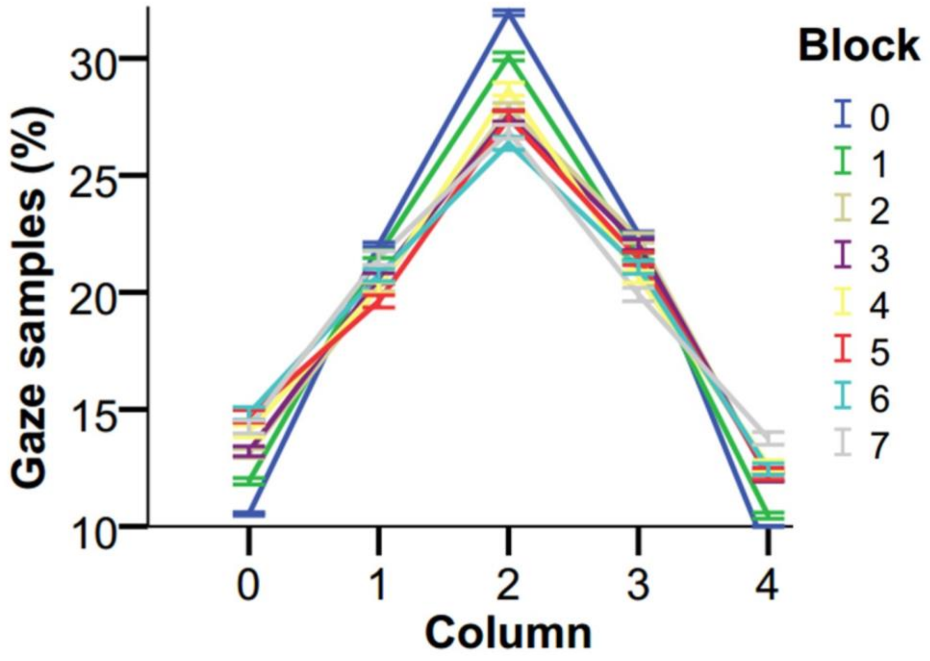


→ Users tend to look at the center.

Results



→ No statistical differences, but some tendency to center.



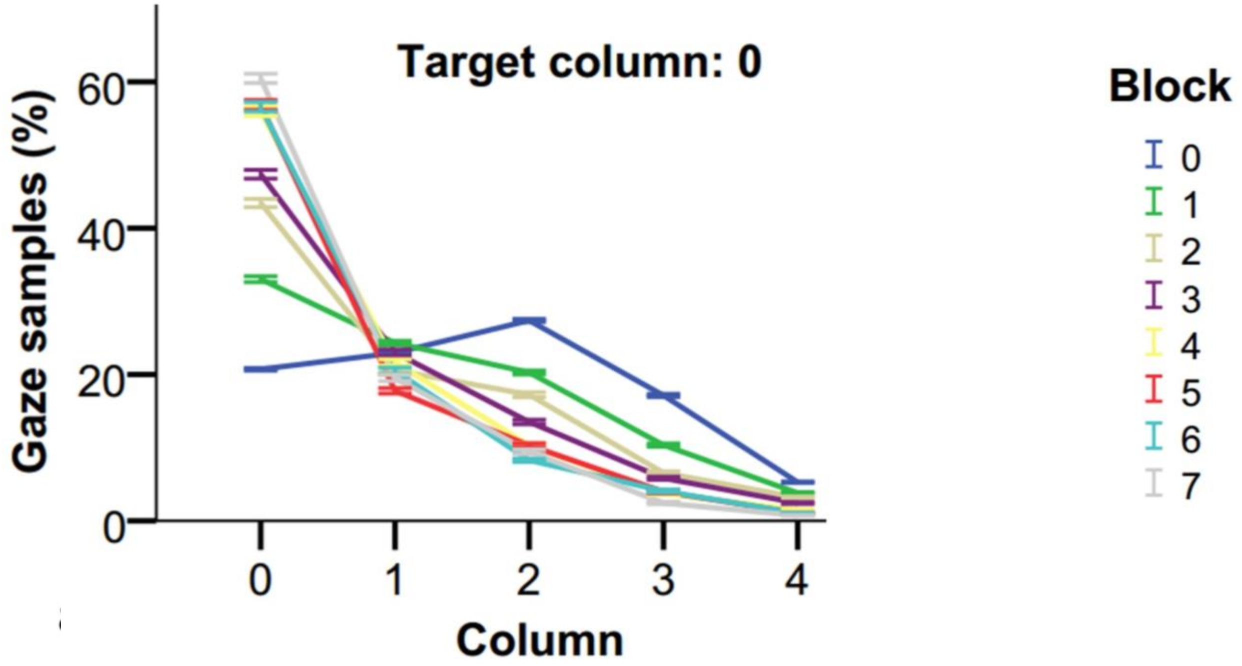
→ Users tend to look at the center.

We model visual search as a linear scan from the center of the columns:

$$T_{vs} = |(colLen/2 - col)| \times T_{col} + v$$

where *v* is the bias term, and T_{col} is the time for the user to visually scan each column.

Results



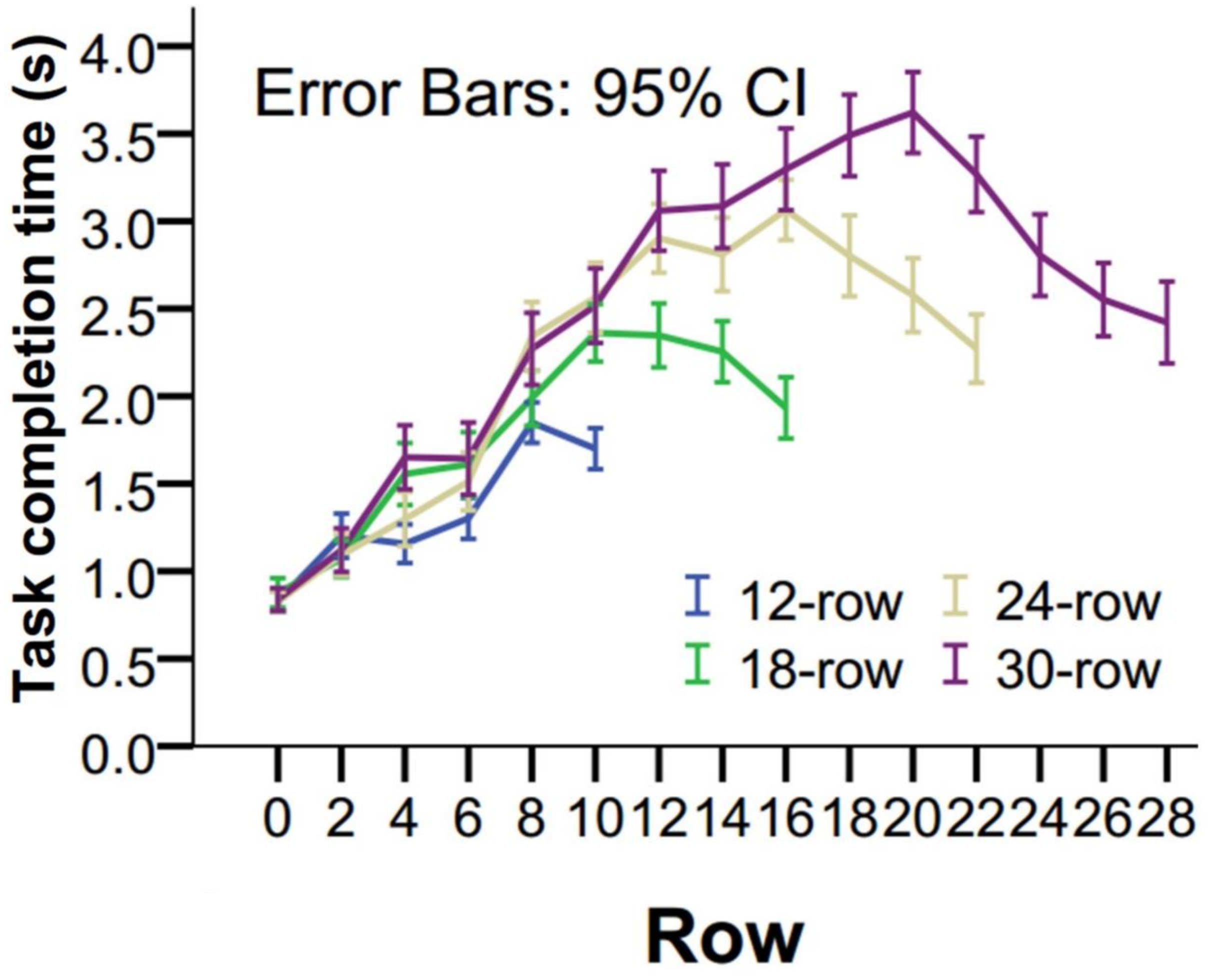
→With experience, users look closer to the target

T_{row} incorporates the learning rate that decreases logarithmically with experience, modeled by the power law of practice:

$$T_{row} = a_r \times e^{(-b_r \times t)} + c_r$$

where t denotes the number of previous trials, and a_r, b_r, and c_r are parameters to be learned.

Results

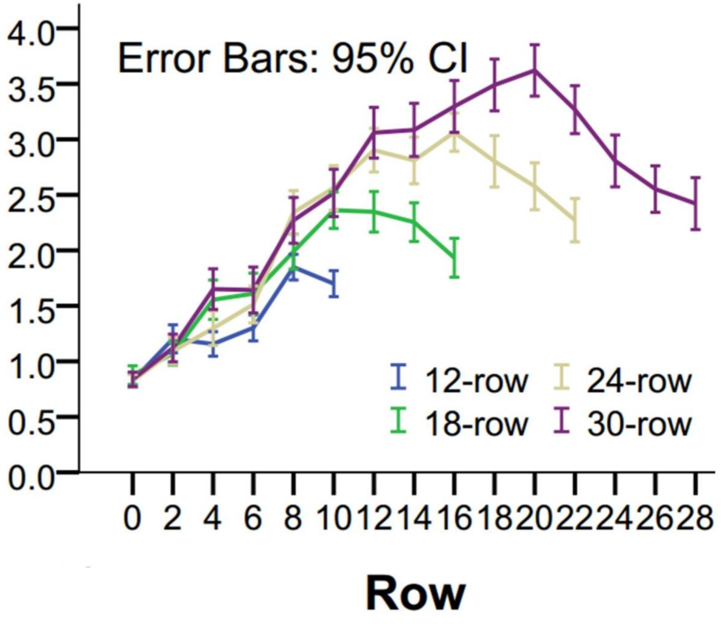


- Significant statistical differences
- Time initially increases
- Time decreases toward end
- Why?

Results

Learning | Visual search | Navigation | Pointing

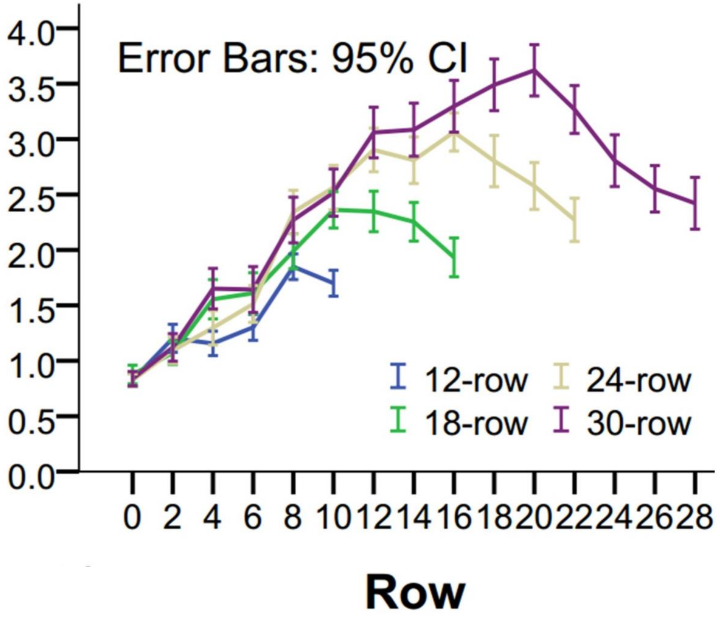
Question: Time initially increases but decreases towards end – why?



Results

Learning | Visual search | Navigation | Pointing

Question: Time initially increases but decreases towards end – why?



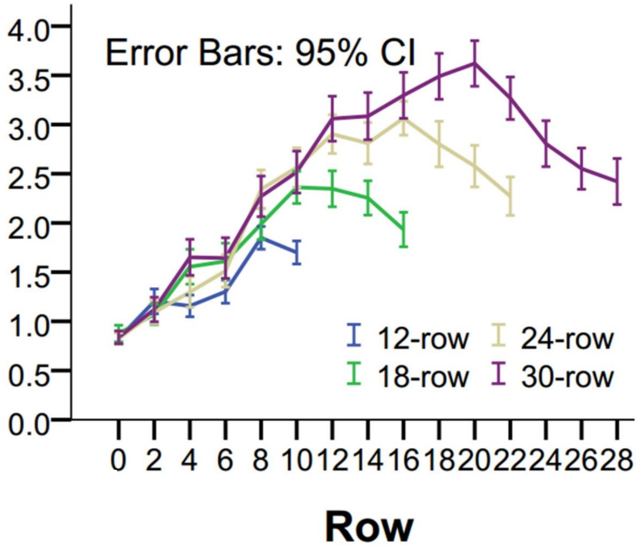
Top-down (80.2%): The user navigates from the top of the menu continuously downwards, until the target is found.

Bottom-up (19.8%): The user performs a flick gesture to absolutely scroll to the bottom. Then, the user selects a target (17.3%), or navigates up and selects another (2.5%).

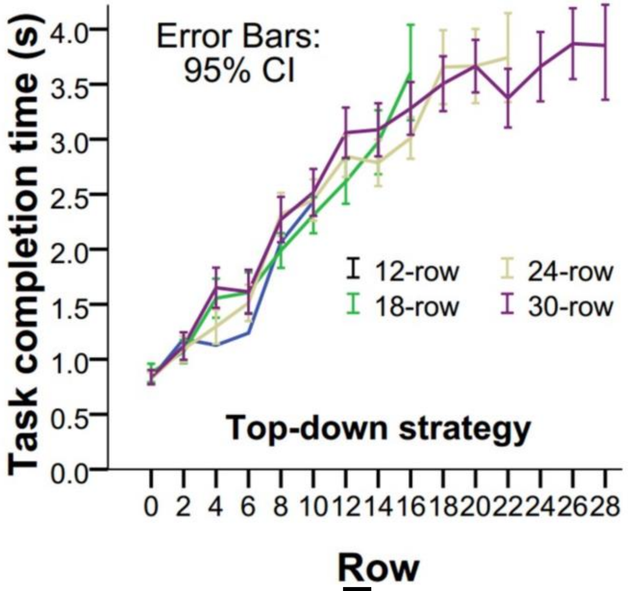
Results

Learning | Visual search | Navigation | Pointing

Question: Time initially increases but decreases towards end – why?

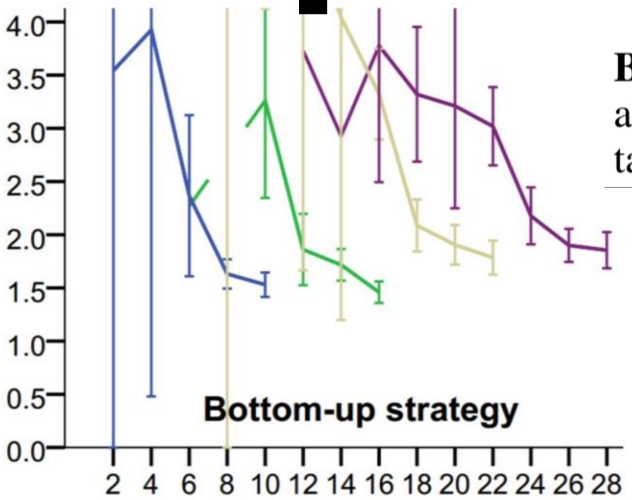


=



Top-down (80.2%): The user navigates from the top of the menu continuously downwards, until the target is found.

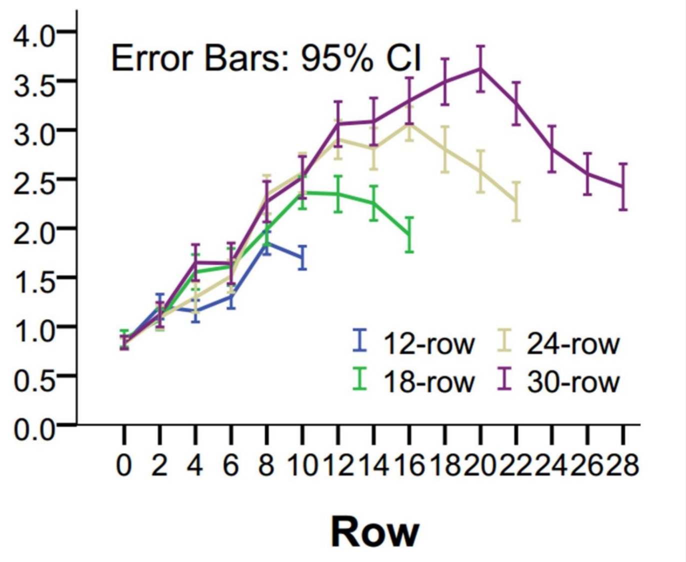
+



Bottom-up (19.8%): The user performs a flick gesture to absolutely scroll to the bottom. Then, the user selects a target (17.3%), or navigates up and selects another (2.5%).

Results

Learning | Visual search | Navigation | Pointing



Probabilistic strategy regulation:

$$T_{nav} = (1-s) \times Strat_{top} + s \times Strat_{bot}$$

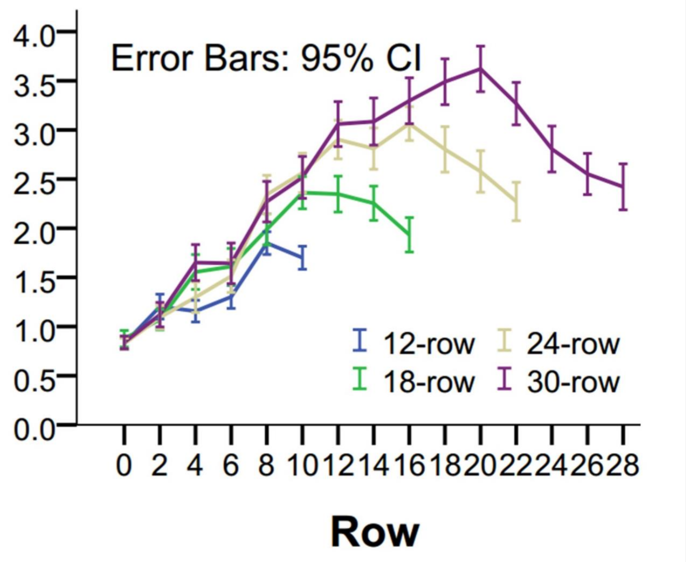
For each strategy, time is linear with row position:

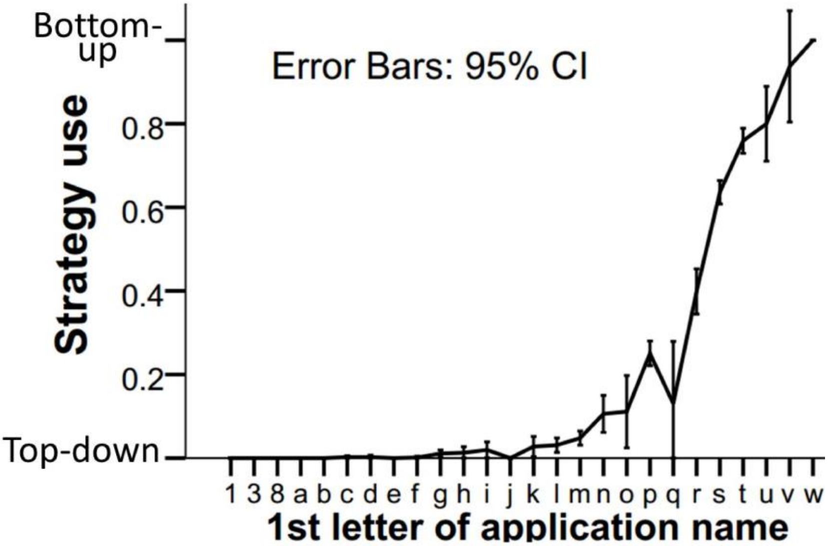
$$Strat_{top} = pos_{row} \times T_{row} + b_{top}$$

$$Strat_{bot} = (len_{row} - pos_{row}) \times T_{row} + b_{bot}$$

Results

Learning | Visual search | Navigation | Pointing





Probabilistic strategy regulation:

$$T_{nav} = (1 - s) \times Strat_{top} + s \times Strat_{bot}$$

For each strategy, time is linear with row position:

$$Strat_{top} = pos_{row} \times T_{row} + b_{top}$$

$$Strat_{bot} = (len_{row} - pos_{row}) \times T_{row} + b_{bot}$$

s_{prob} is a sigmoidal function that outputs a probability between 0 and 1, based on a linear combination of three values: the first letter of the target name, the user experience, and the gridlength:

$$s_{prob} = sigmoid(s_b + s_{w1} \times len_{row} + s_{w2} \times l + s_{w3} \times s_{exp})$$

where s_b and s_{wi} are the bias and weights, and s_{exp}, the expertise of using a strategy

Results

Pointing modelled by Fitts' Law

$$T = a + b \log_2 \left(1 + \frac{D}{W}\right)$$

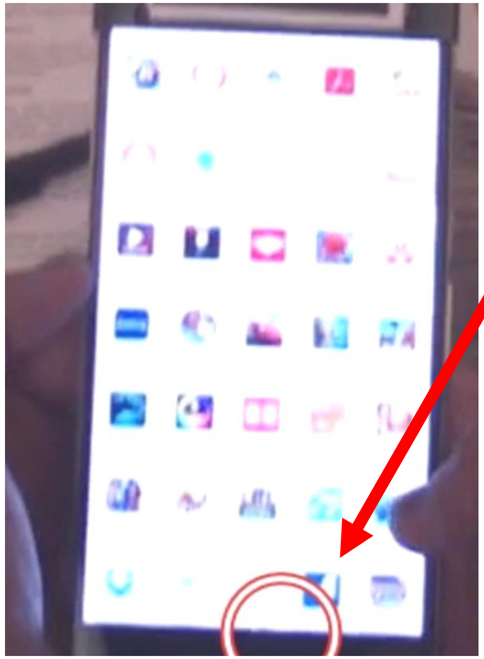
D = distance (touch_start, target)

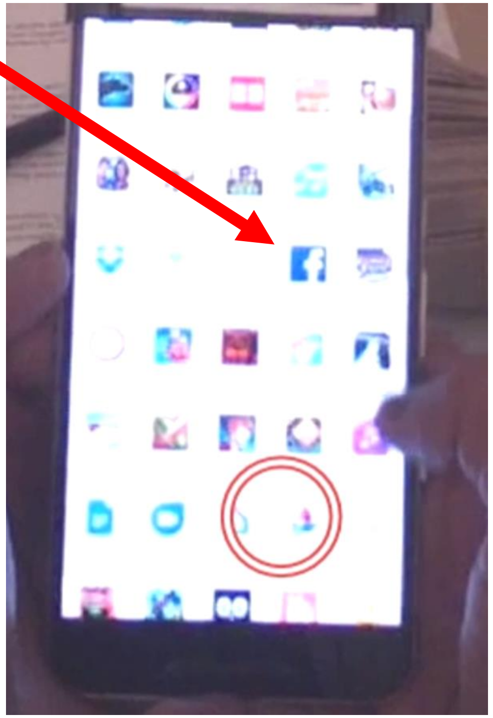
- Touch_start: modelled as center of screen
- Target:
  - X: given by column
  - Y: unknown

Question: How to acquire Y position?

Results

Where is Facebook?

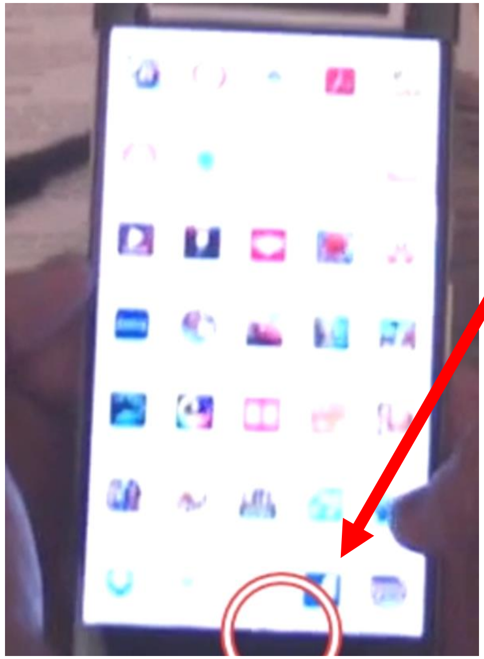


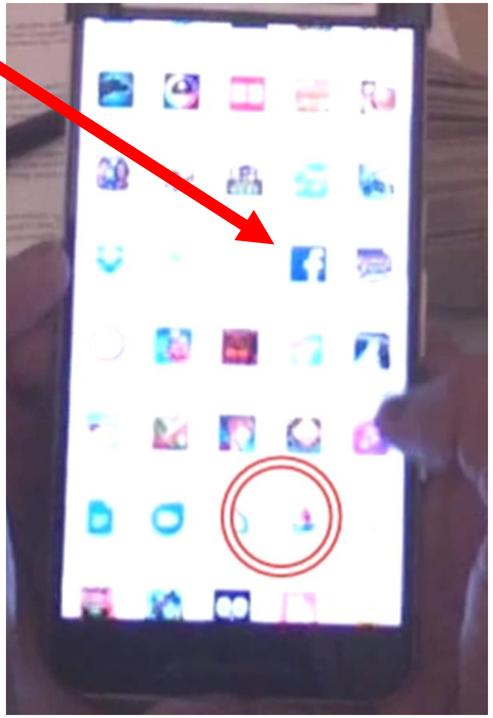


Results

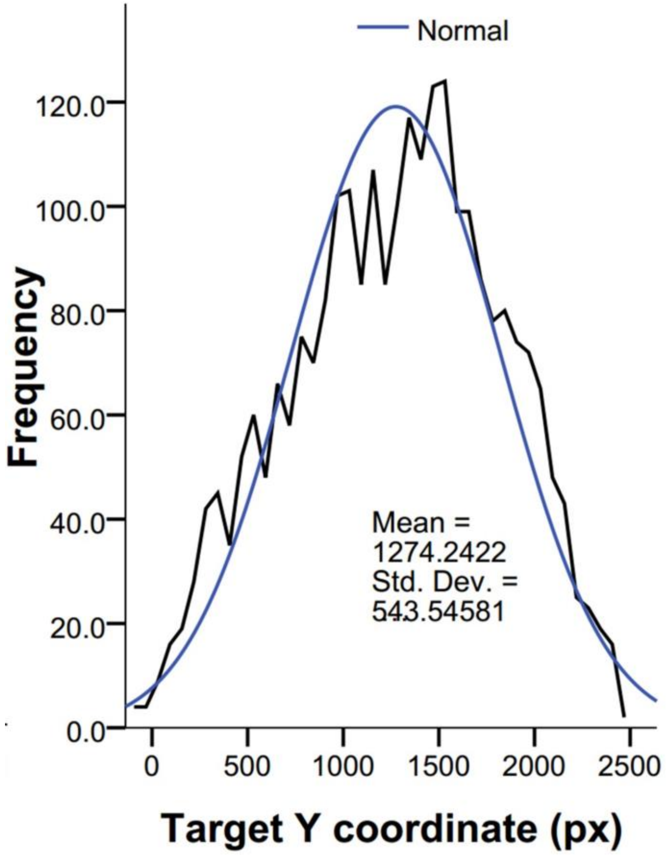
Learning | Visual search | Navigation | Pointing

Where is Facebook?





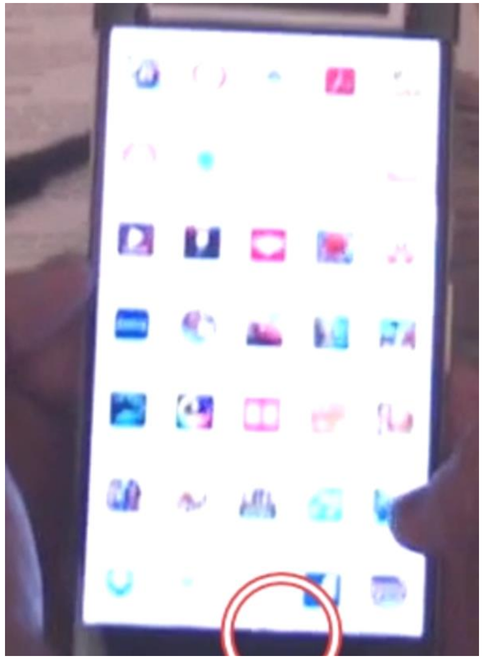
Where was the target on average?



Results

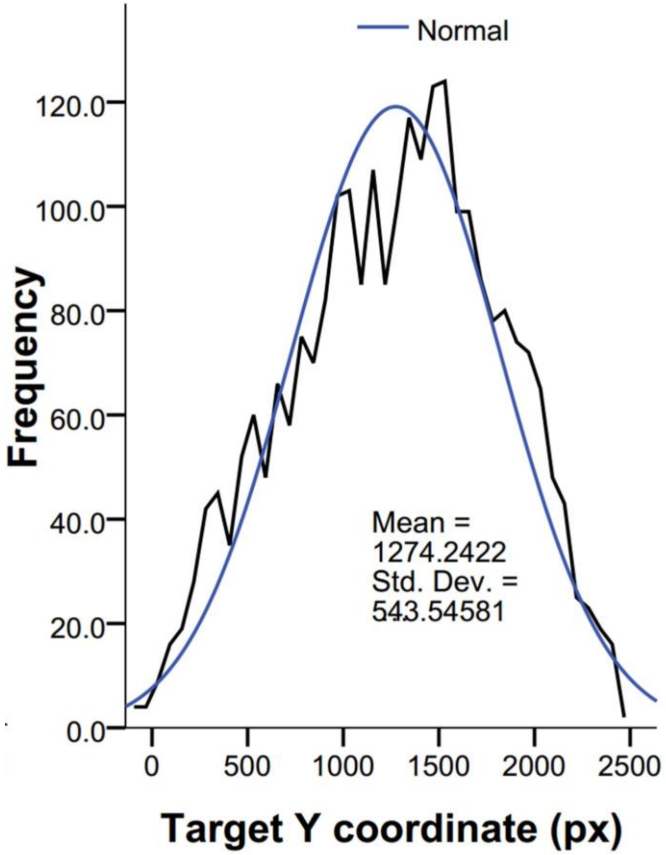
Learning | Visual search | Navigation | Pointing

How to model?



Each row gets a probability:
→ 0.05
→ 0.08
→ 0.1
→ 0.15
→ 0.19
→ 0.13
→ 0.11
→ 0.09
→ 0.02

Where was the target on average?



Results

Learning | Visual search | Navigation | Pointing

We compute the weighted average of the cost for each row j to estimate pointing time:

$$T_{point} = \sum_{j=1}^{view_{rows}} p_{row_j} \times T_{point_{row_j}}$$

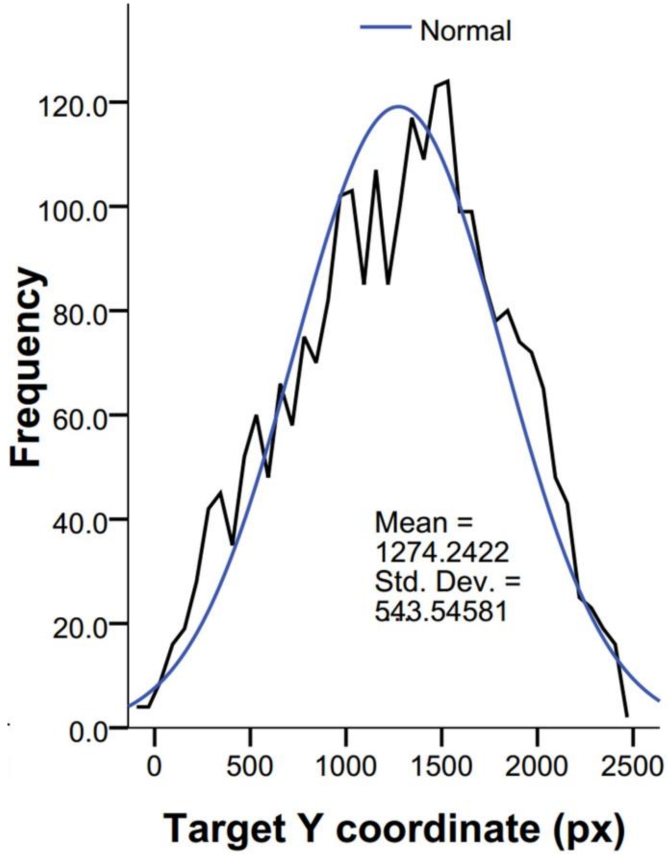
For each row j, time is calculated by the Fitts' Law model:

$$T_{point_{row_j}} = a_f + b_f \log_2\left(1 + \frac{\mathrm{d}((pos_{col}, row_j), cen)}{W}\right)$$

The probability for the target to be on each row j is determined by a probability density of normal distribution to reflect how the Y positions are distributed across the screen in our study:

$$p_{row_j} = \frac{1}{\sigma\sqrt{2\pi}} e^{-(row_j/view_{row}-\mu)^2/2\sigma^2}$$

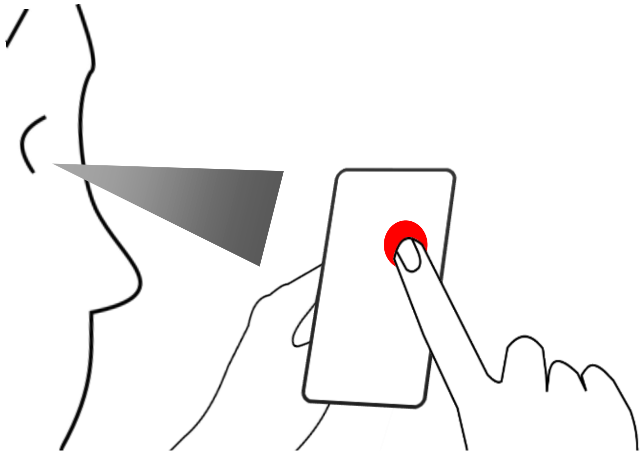
Where was the target on average?



Project 1: User Performance Modelling

Implicit

Data collection & offline analysis



Outline:

1. Idea
2. User study
3. Results
4. **Model & Evaluation**

Model & Evaluation

$$T_i = T_{nav} + T_{vs} + T_{point}$$

$$T_{nav} = (1-s) \times Strat_{top} + s \times Strat_{bot}$$
$$s_{prob} = sigmoid(s_b + s_{w1} \times len_{row} + s_{w2} \times l + s_{w3} \times s_{exp})$$
$$Strat_{top} = pos_{row} \times T_{row} + b_{top}$$
$$Strat_{bot} = (len_{row} - pos_{row}) \times T_{row} + b_{bot}$$
$$T_{row} = a_r \times e^{(-b_r \times t)} + c_r$$

$$T_{vs} = |(colLen/2 - col)| \times T_{col} + v$$
$$T_{col} = a_{vs} \times e^{(-b_{vs} \times t)} + c_{vs}$$

$$T_{point} = \sum_{j=1}^{view_{rows}} p_{row_j} \times T_{point_{row_j}}$$
$$T_{point_{row_j}} = a_f + b_f \log_2\left(1 + \frac{\mathrm{d}((pos_{col}, row_j), cen)}{W}\right)$$
$$p_{row_j} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(row_j/view_{row} - \mu\right)^2 / 2\sigma^2}$$

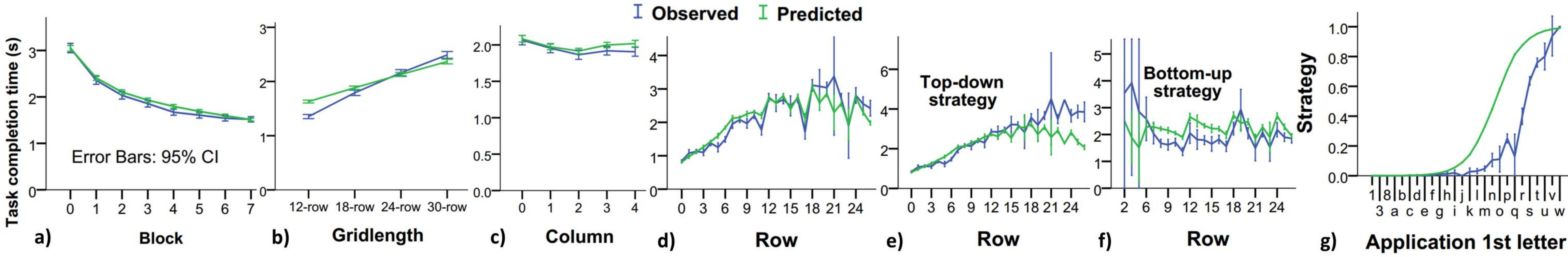
Model & **Evaluation**

Evaluation details:

- Model implemented in TensorFlow with stochastic gradient descent
- Trained on the study data
- 2-fold cross-validation
- Model fitting: R^2 between 0 (no fit) and 1 (same data)

Results:

Block: $R^2 = .990$ (8 blocks)
Block×Gridlength: $R^2 = .942$ (8 block×4 grid)
Column×Gridlength: $R^2 = .909$ (5 col×4 grid)
Row×Gridlength: $R^2 = .813$ (6+9+12+15 rows)



Prediction benefit depends on interaction cost.

→ Incorporate interaction cost in prediction.

→ Use a model that predicts cost, i.e. app selection time.

→ What model? $T_i = T_{nav} + T_{vs} + T_{point}$

Model integration

"Normal" probability based optimisation
(cost of selecting an app in drawer)

$$cost_t^i = \begin{cases} C & \text{if } i \in Top5(P_t) \\ G(i,t,g) & \text{otherwise} \end{cases}$$

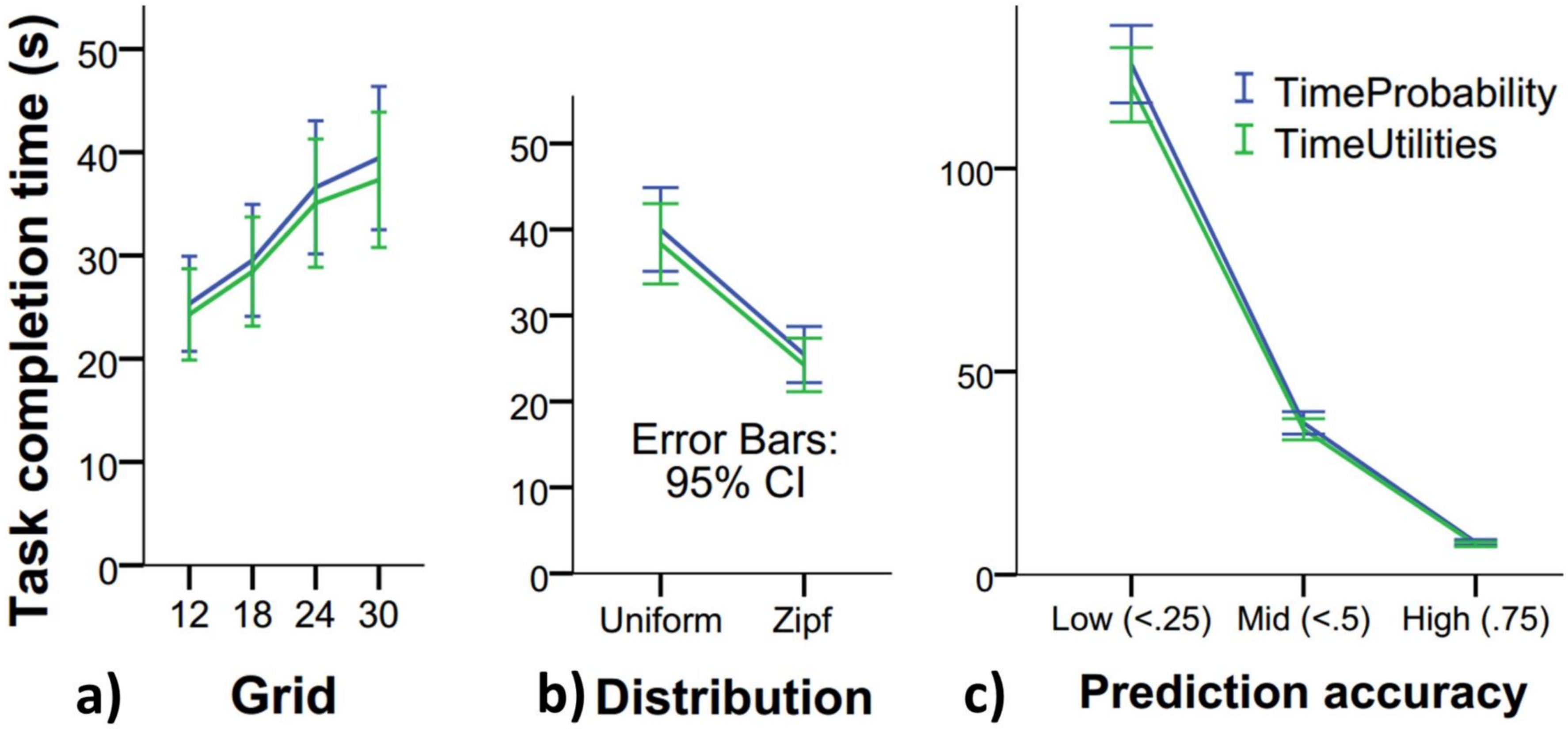
New utility based optimisation
(G represents the model)

$$U_t = P_t \odot G(t,g)$$

New optimization based on utility

$$cost_t^i = \begin{cases} C & \text{if } i \in Top5(U_t) \\ G(i,t,g) & \text{otherwise} \end{cases}$$

Simulation experiment

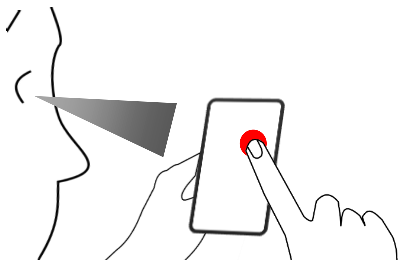


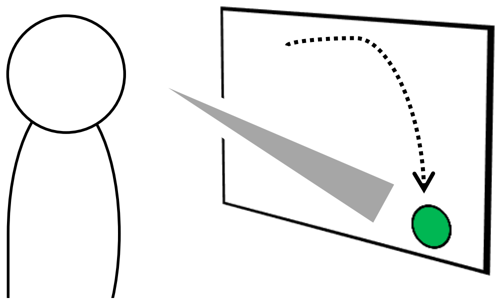
Visual attention

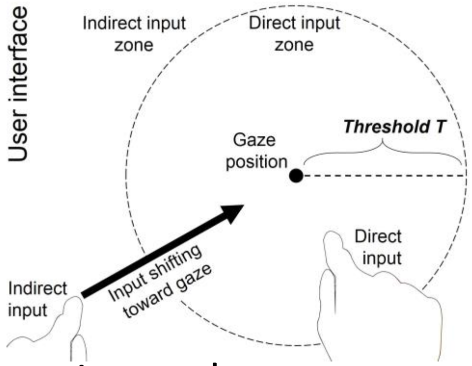
Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

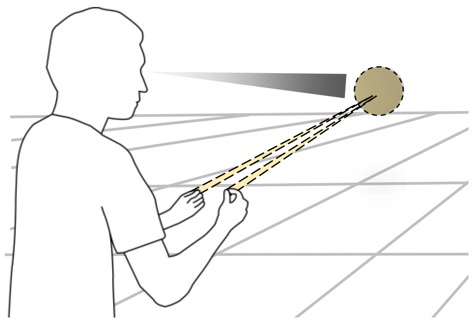
Implicit ⟷ Explicit


User performance modelling

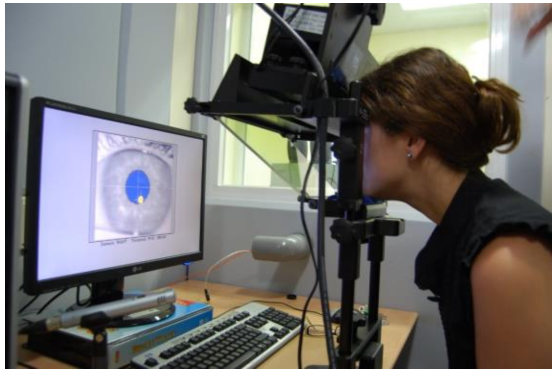

Movement correlation & calibration

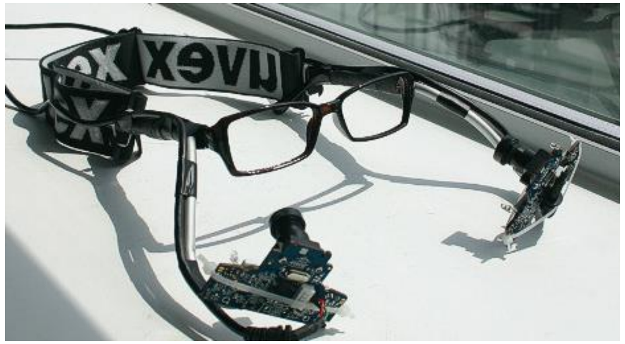


Input shortcuts


Gaze + Manual Input

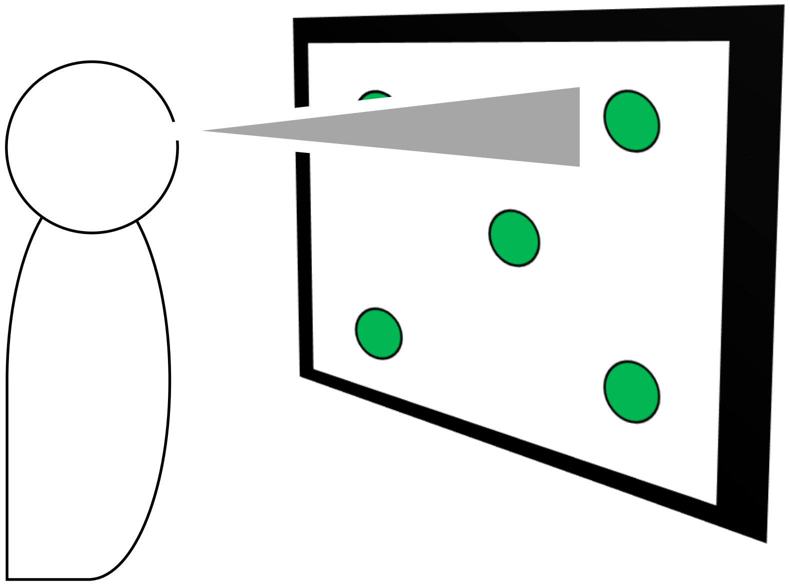
Eye trackers require calibration





Motivation: the typical gaze calibration

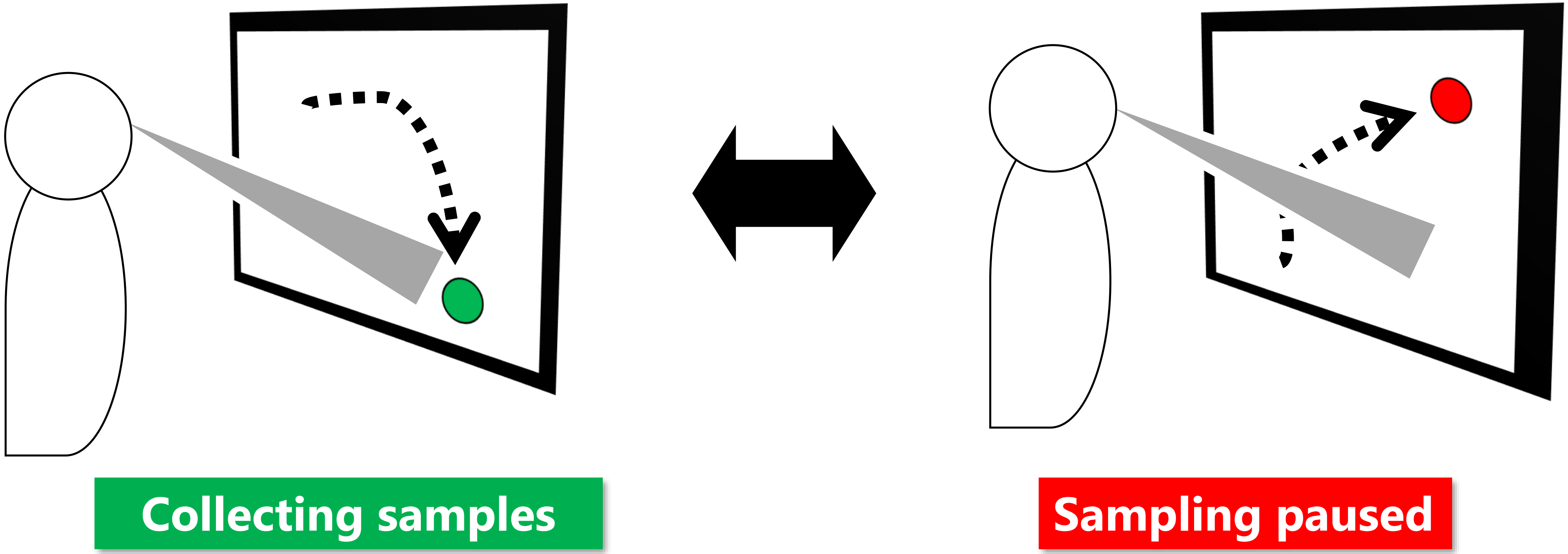
- Establishes mapping between eye input space and screen output space.
- Sampling of eye gaze at known points on-screen.



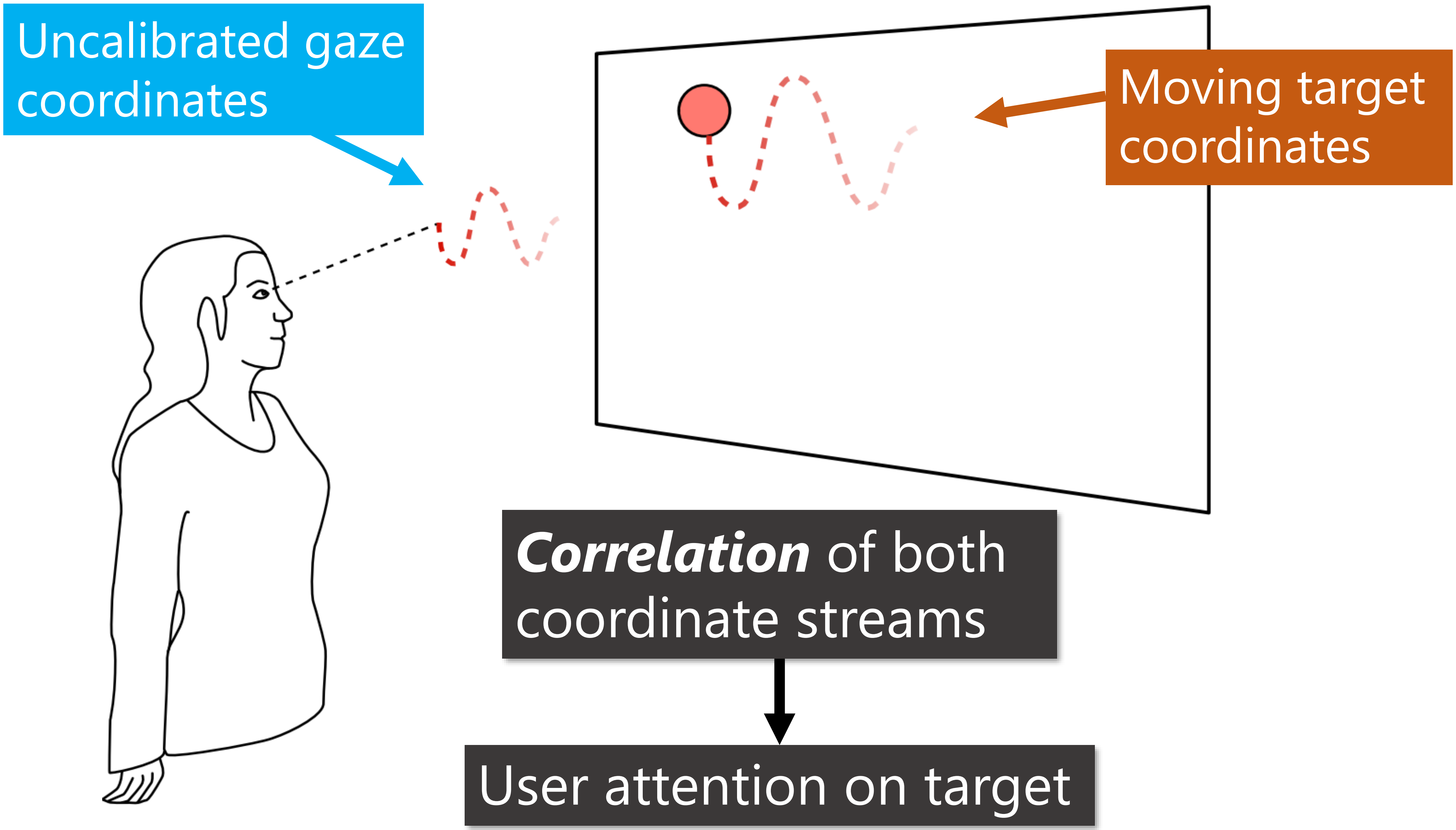
- Tedious, unnatural procedure
- Fixed start and end point
- Reliance on user performance

Pursuit Calibration – a new gaze calibration method

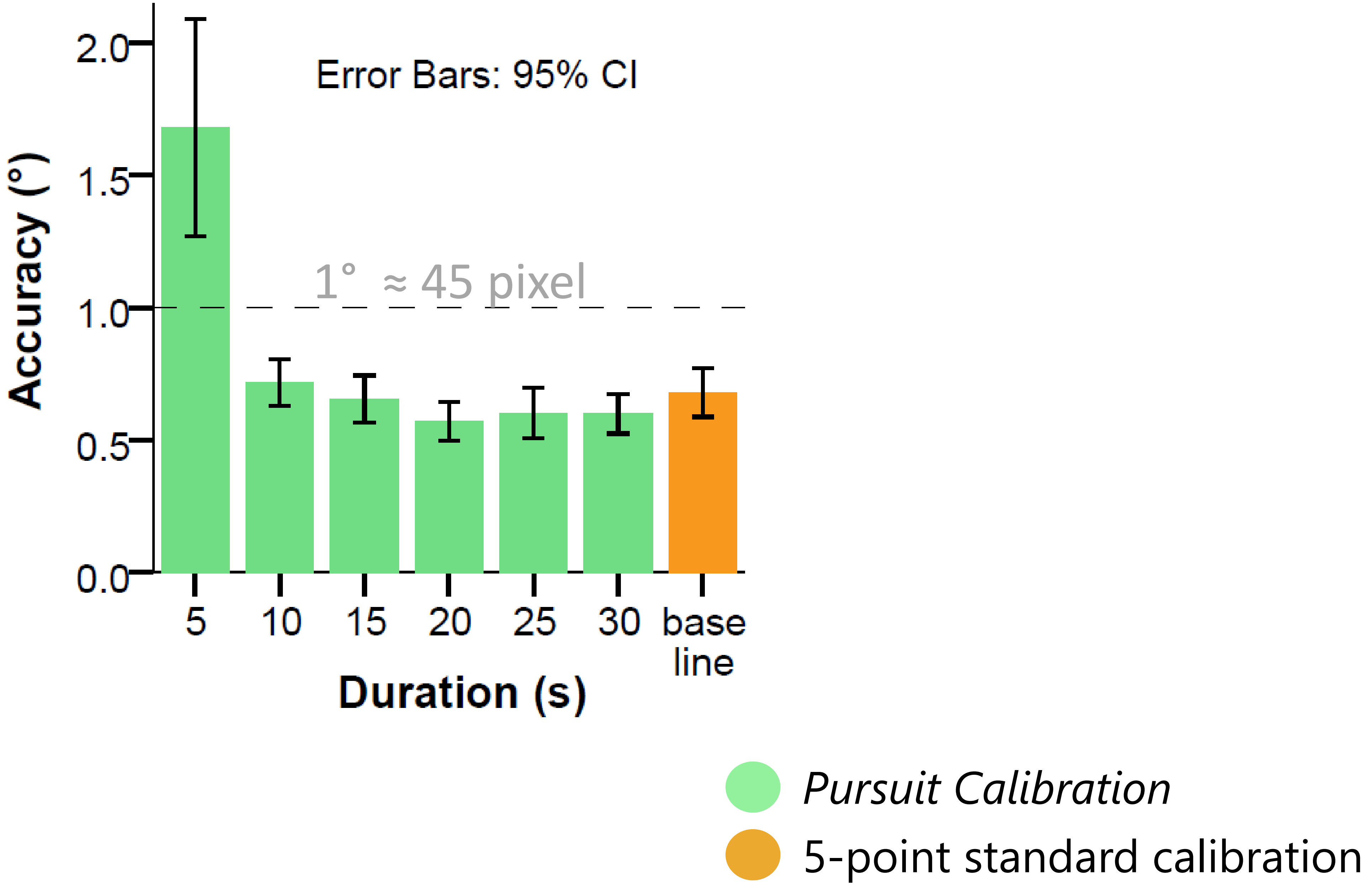
- Based on a moving calibration target.
- Collects calibration samples when the user pays attention to the moving target.

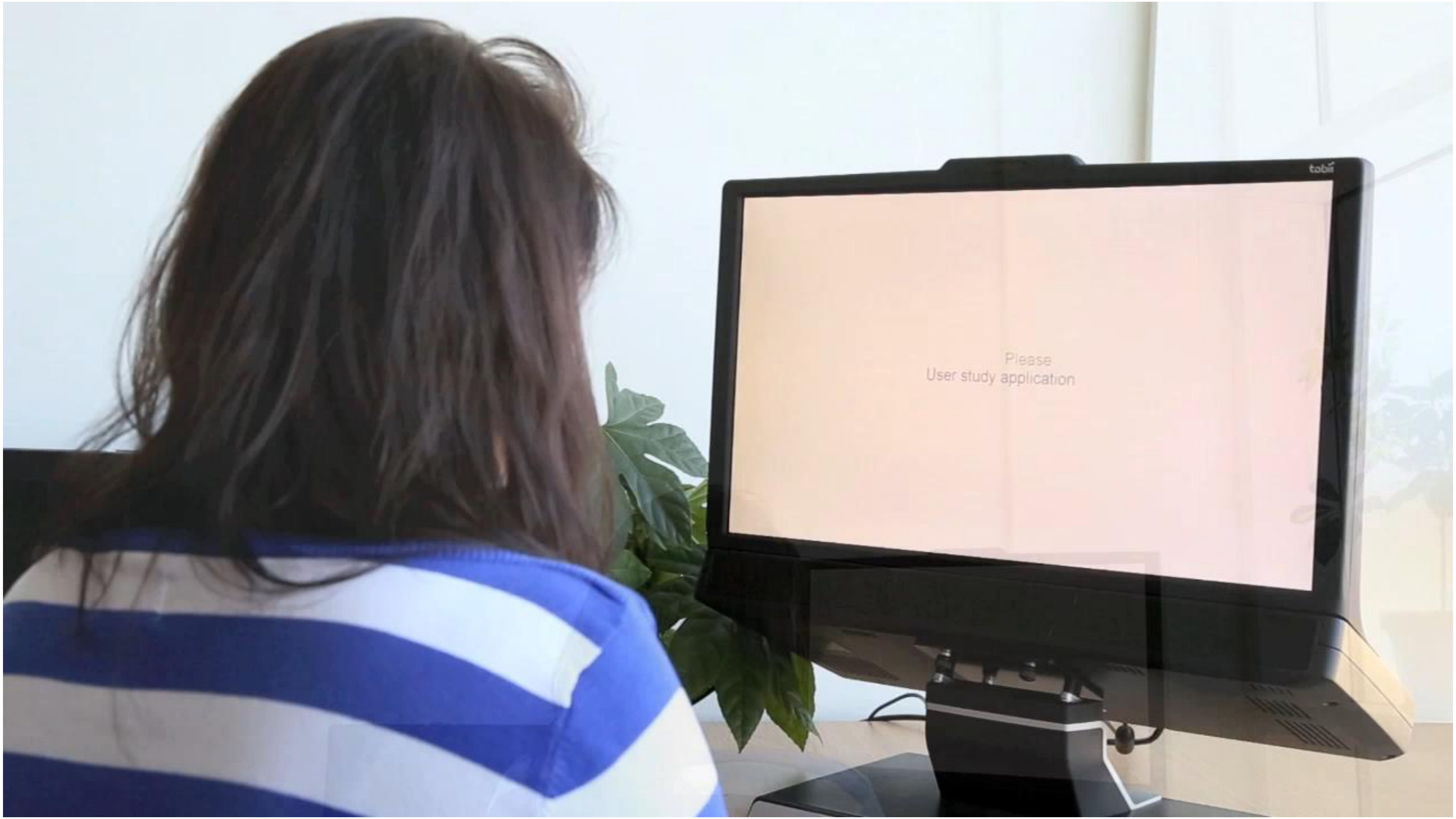



Eyes not detected



Pursuits: Spontaneous Interaction with Displays based on Smooth Pursuit Eye Movement and Moving Targets, M. Vidal, A. Bulling and H. Gellersen, Proc. of UbiComp 2013.


Error Bars: 95% CI
Accuracy (°)
2.0
1.5
1.0
0.5
0.0
1° ≈ 45 pixel
5
10
15
20
25
30
base
line
Duration (s)
Pursuit Calibration
5-point standard calibration

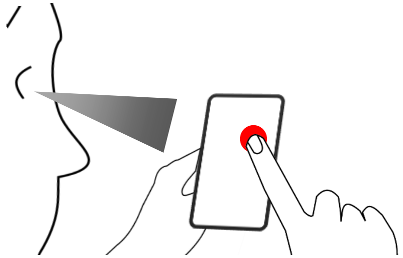

tobii
Please
User study application

Visual attention

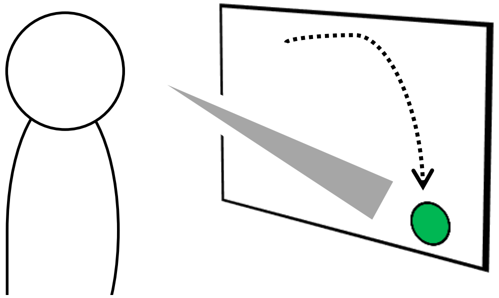
Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

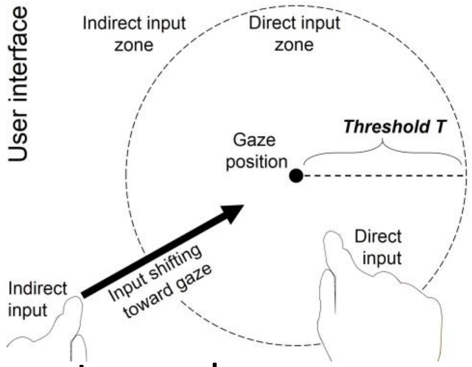
Implicit ⟷ Explicit



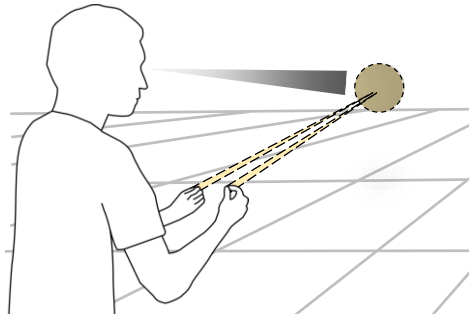
User performance modelling



Movement correlation & calibration



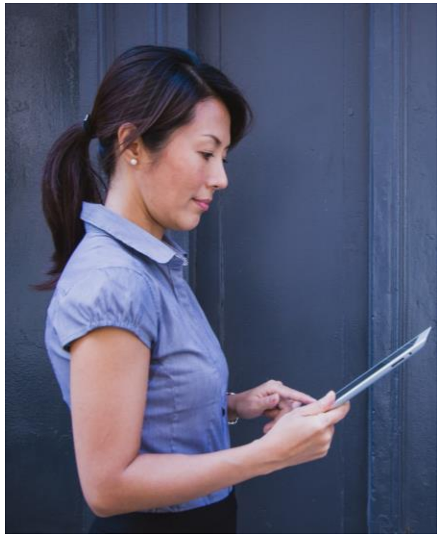
Input shortcuts



Gaze + Manual Input

Devices

Phone



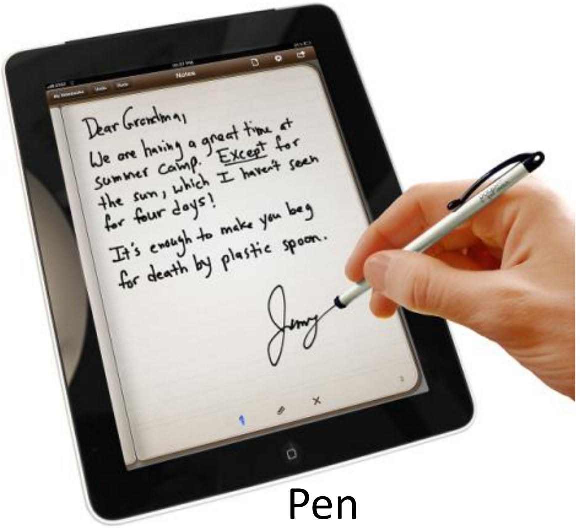
Tablet

Board

VR

Input devices

Touch



Pen

Mouse

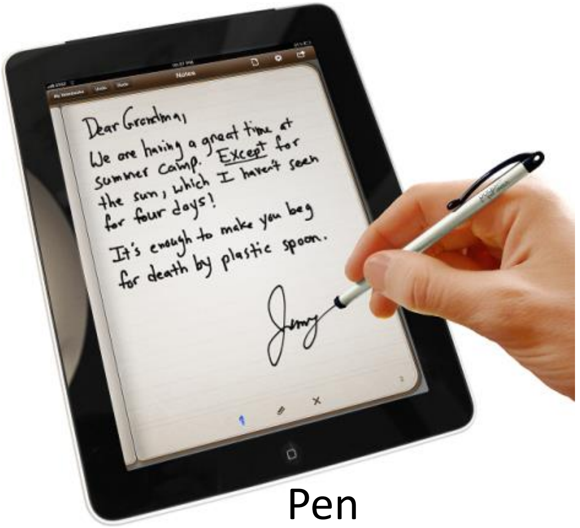
Touchpad

Input devices

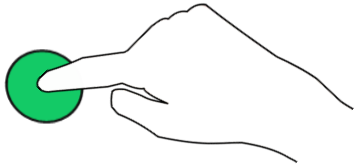
Direct input

Input position equals output position

Touch



Pen

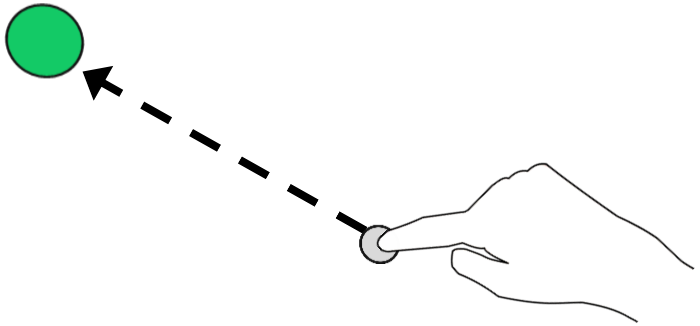


Indirect input

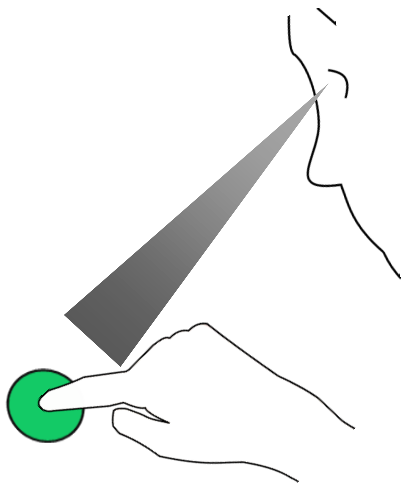
Input is offset from output position

Mouse

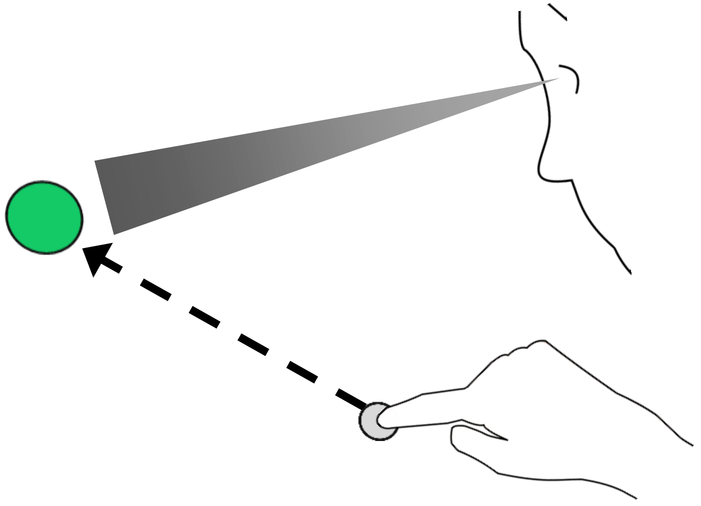
Touchpad



Where are you looking?

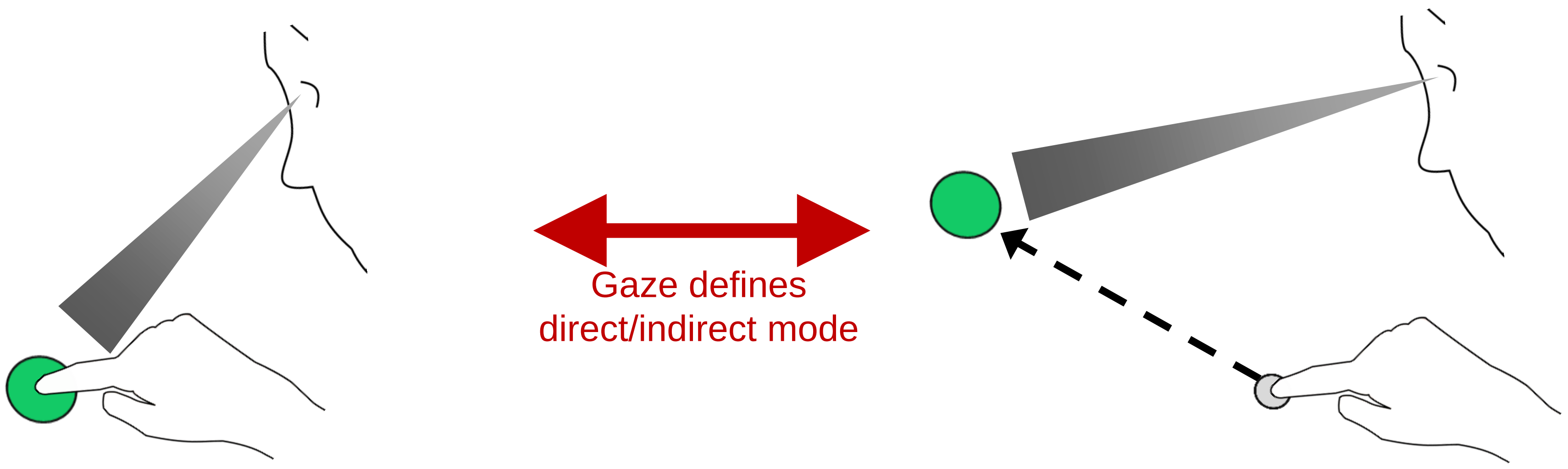


Direct input



Indirect input

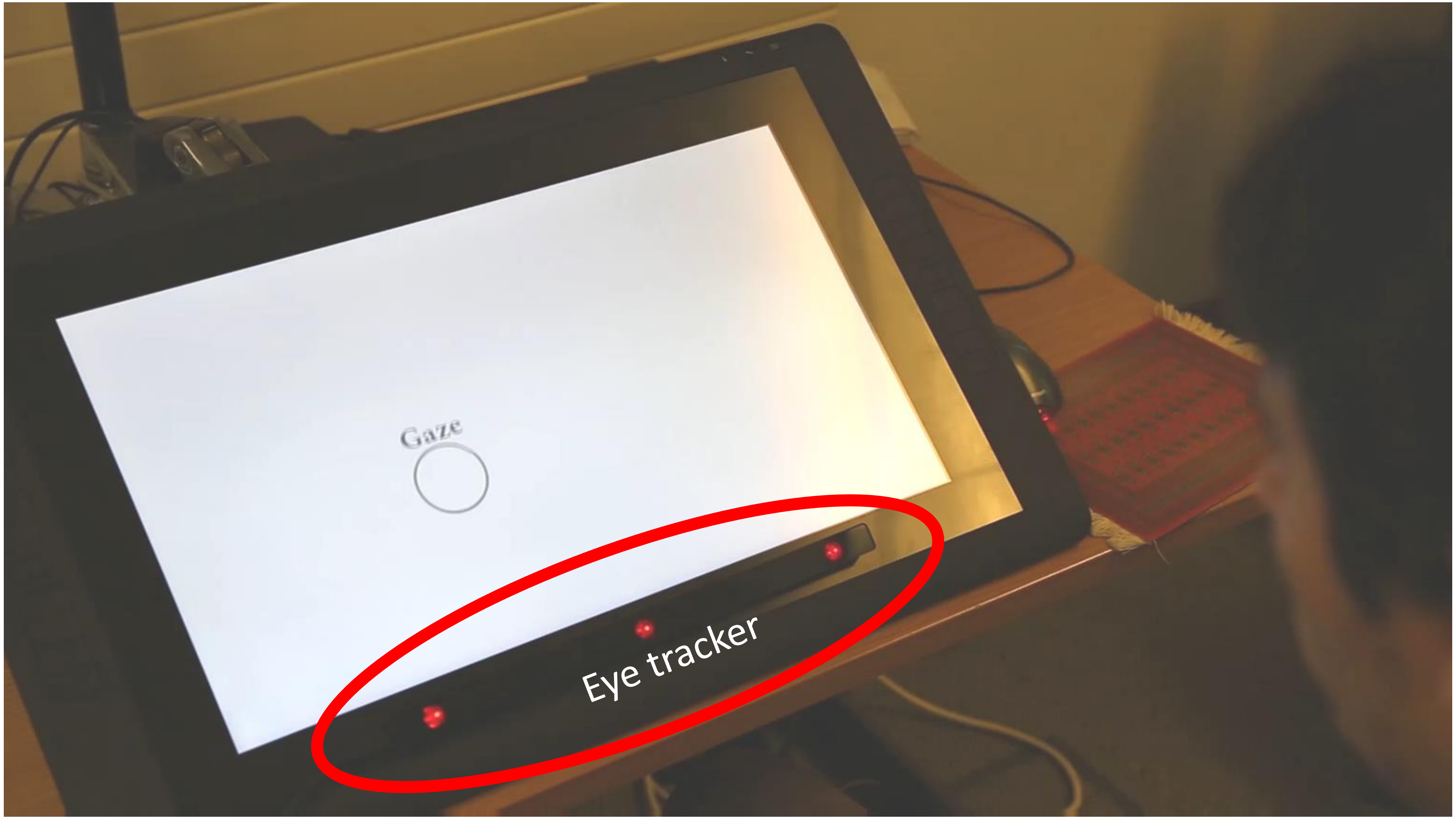
Gaze-shifting



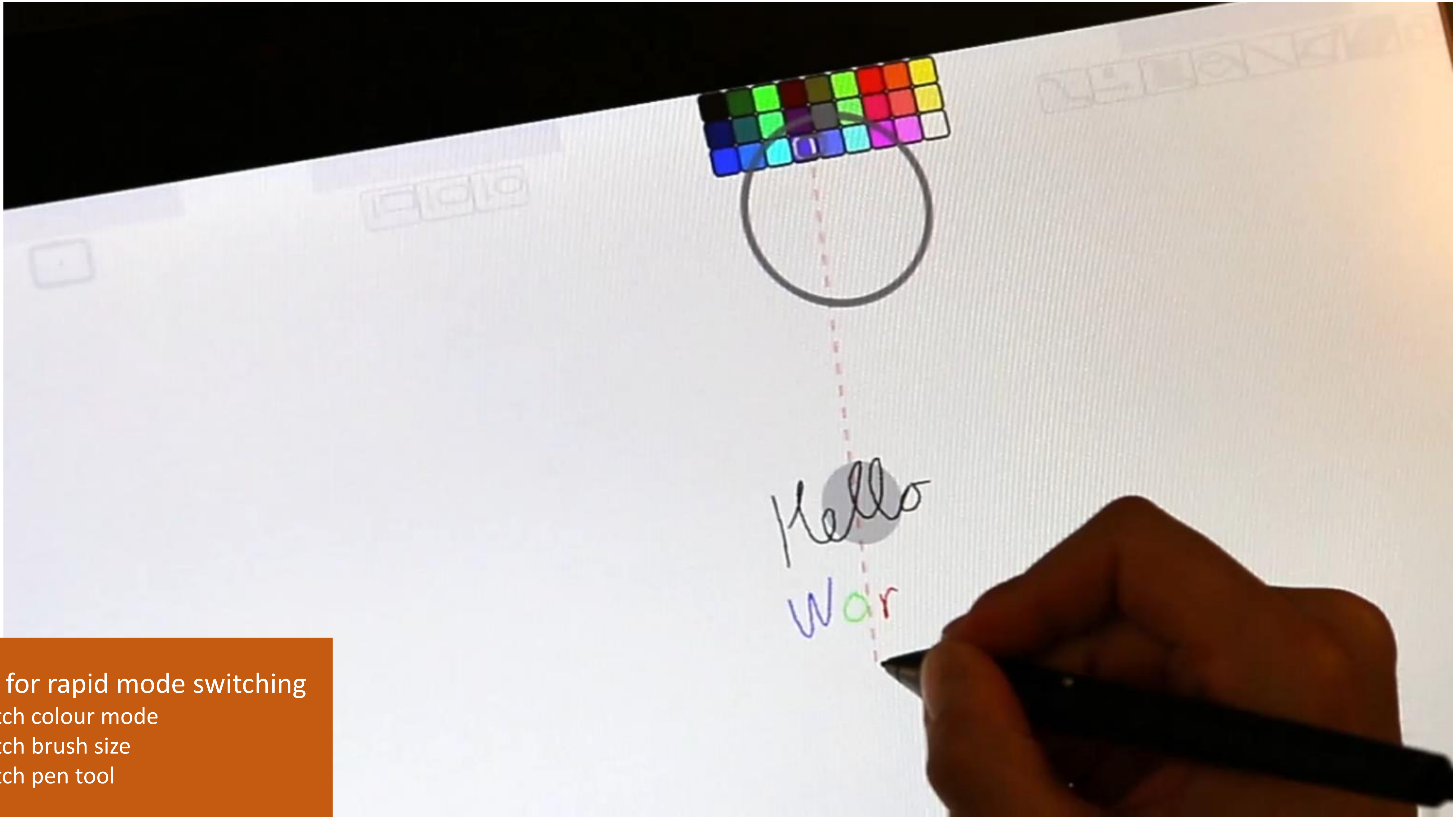
Direct input

Indirect input

Pen and touch display + eye tracking

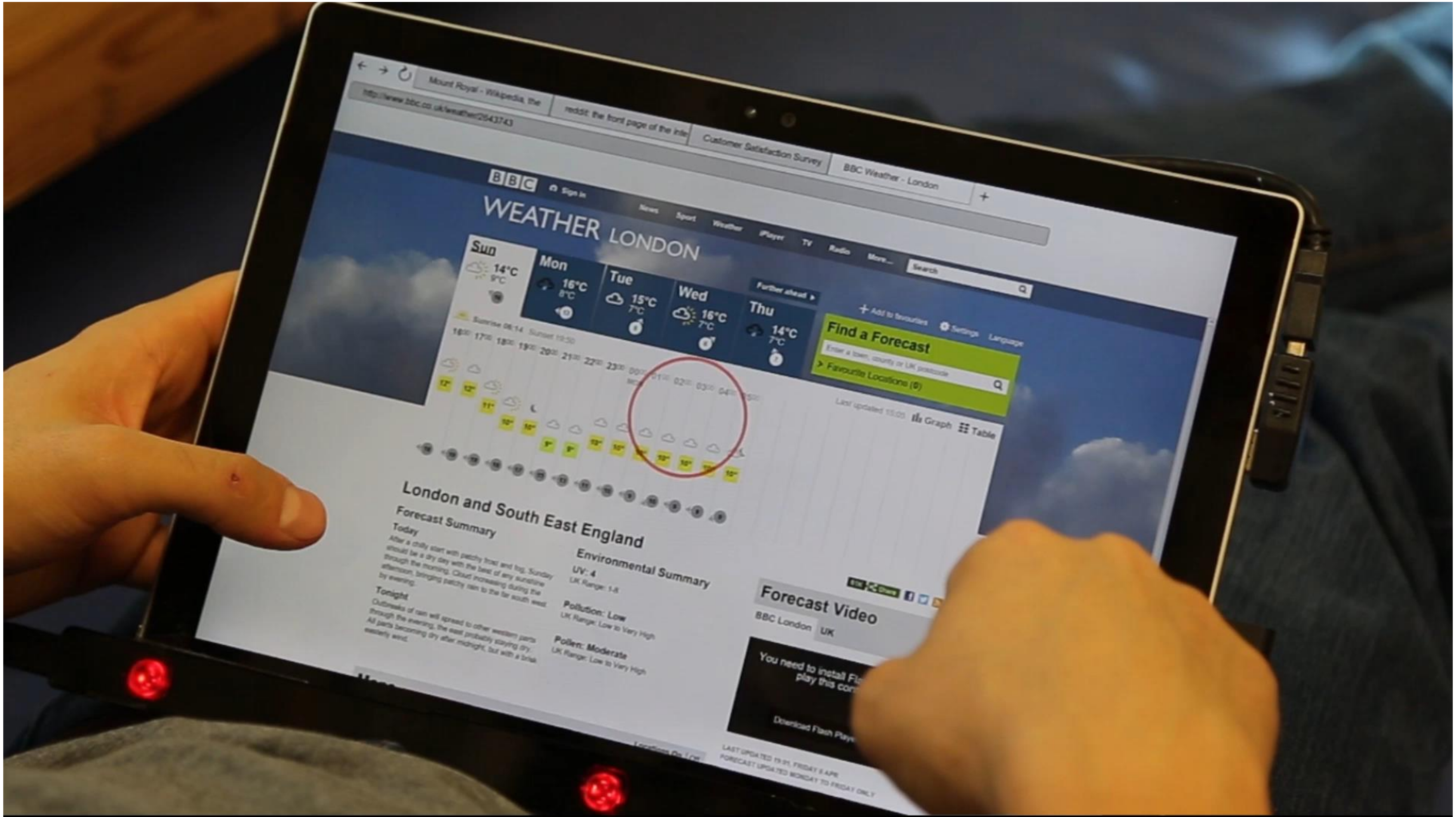


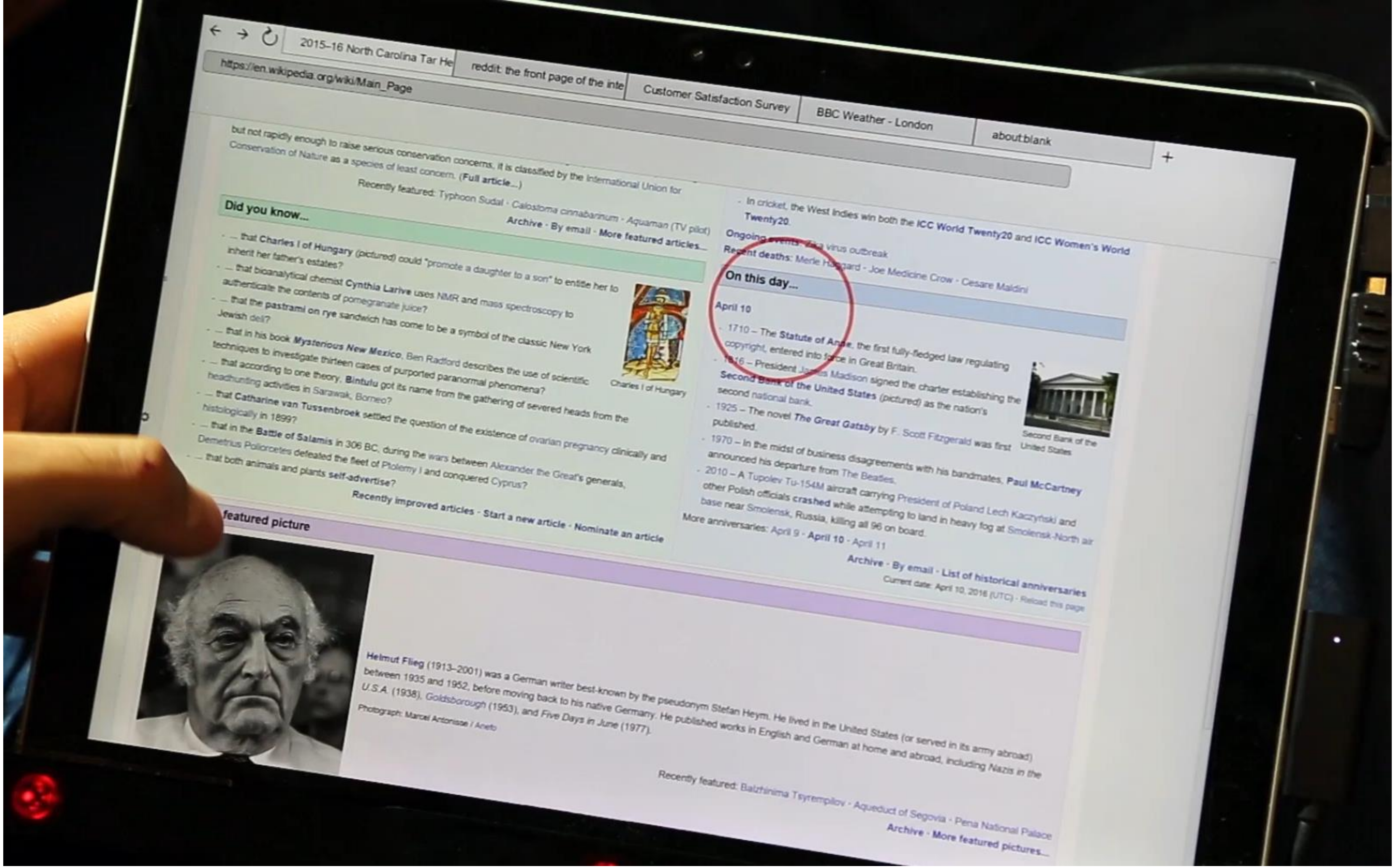
Example application: Multiple menus



Useful for rapid mode switching

- Switch colour mode
- Switch brush size
- Switch pen tool


Mount Royal - Wikipedia, the
reddit: the front page of the inte
Customer Satisfaction Survey
BBC Weather - London
BBC
WEATHER LONDON
Sun
14°C
Mon
16°C
Tue
15°C
Wed
16°C
Thu
14°C
Find a Forecast
Graph
Table
London and South East England
Forecast Summary
Environmental Summary
Forecast Video
BBC London
UK



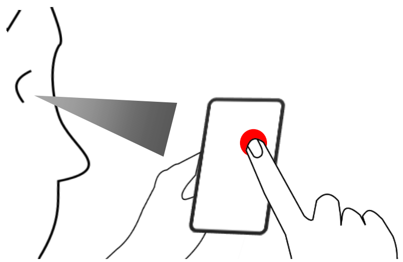
Cursor redirection based on
Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99). ACM, New York, NY, USA, 246-253. DOI=http://dx.doi.org/10.1145/302979.303053

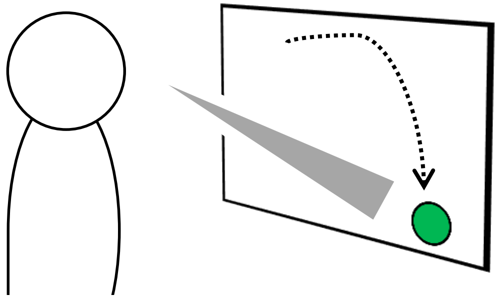
Visual attention

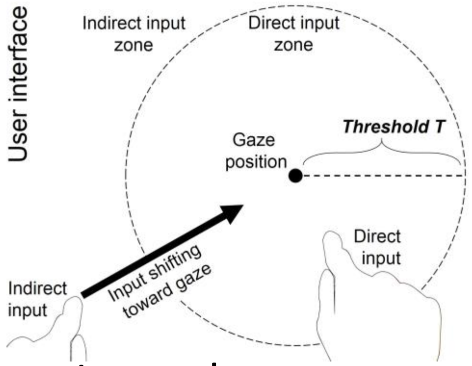
Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

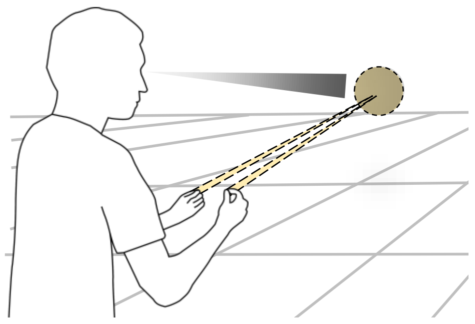
Implicit ⟷ Explicit


User performance modelling


Movement correlation & calibration

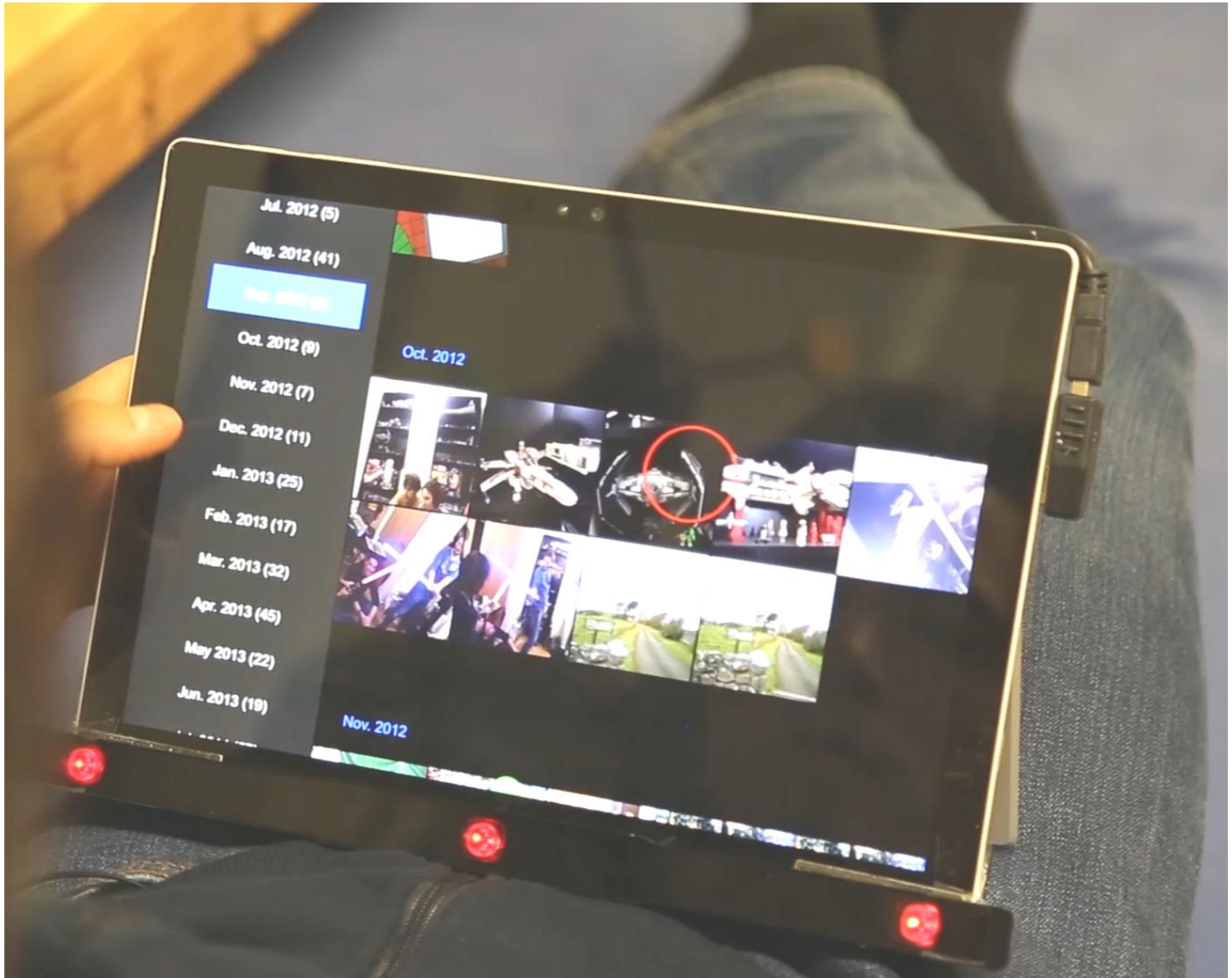


Input shortcuts

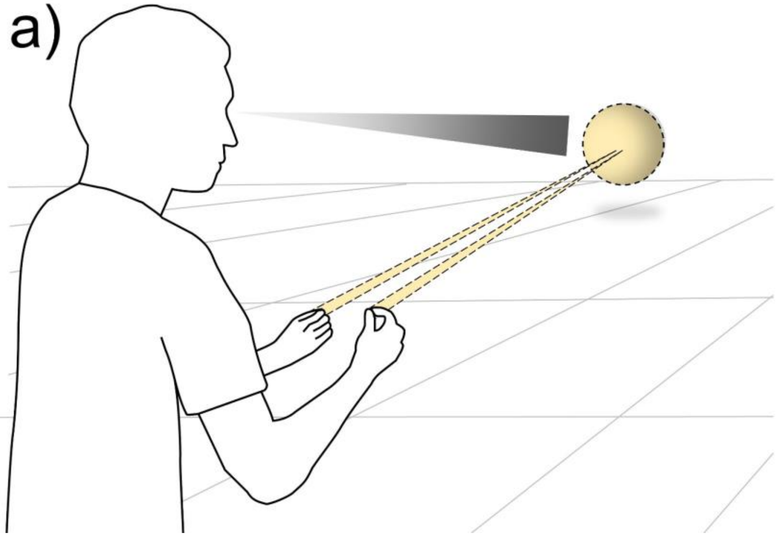

Gaze + Manual Input

Gallery - Scrolling

Gallery – Select image, and back



Gaze + Pinch interaction

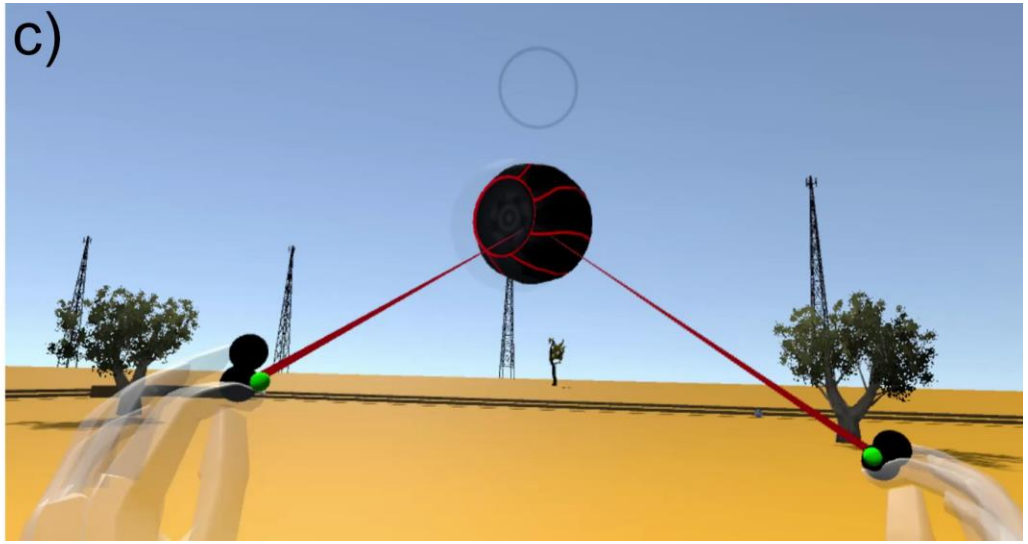


Concept

Gaze selects, hands manipulate

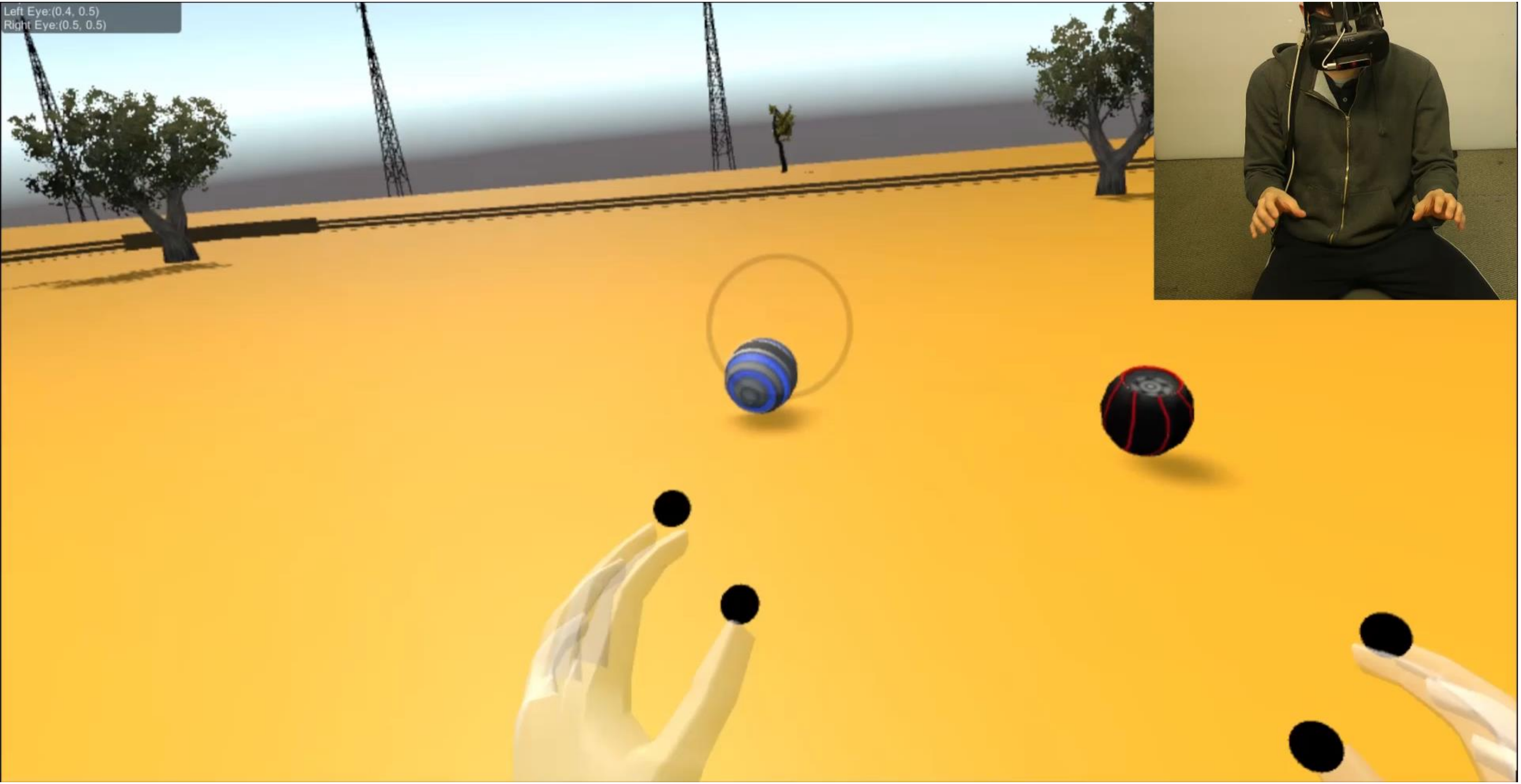
Real

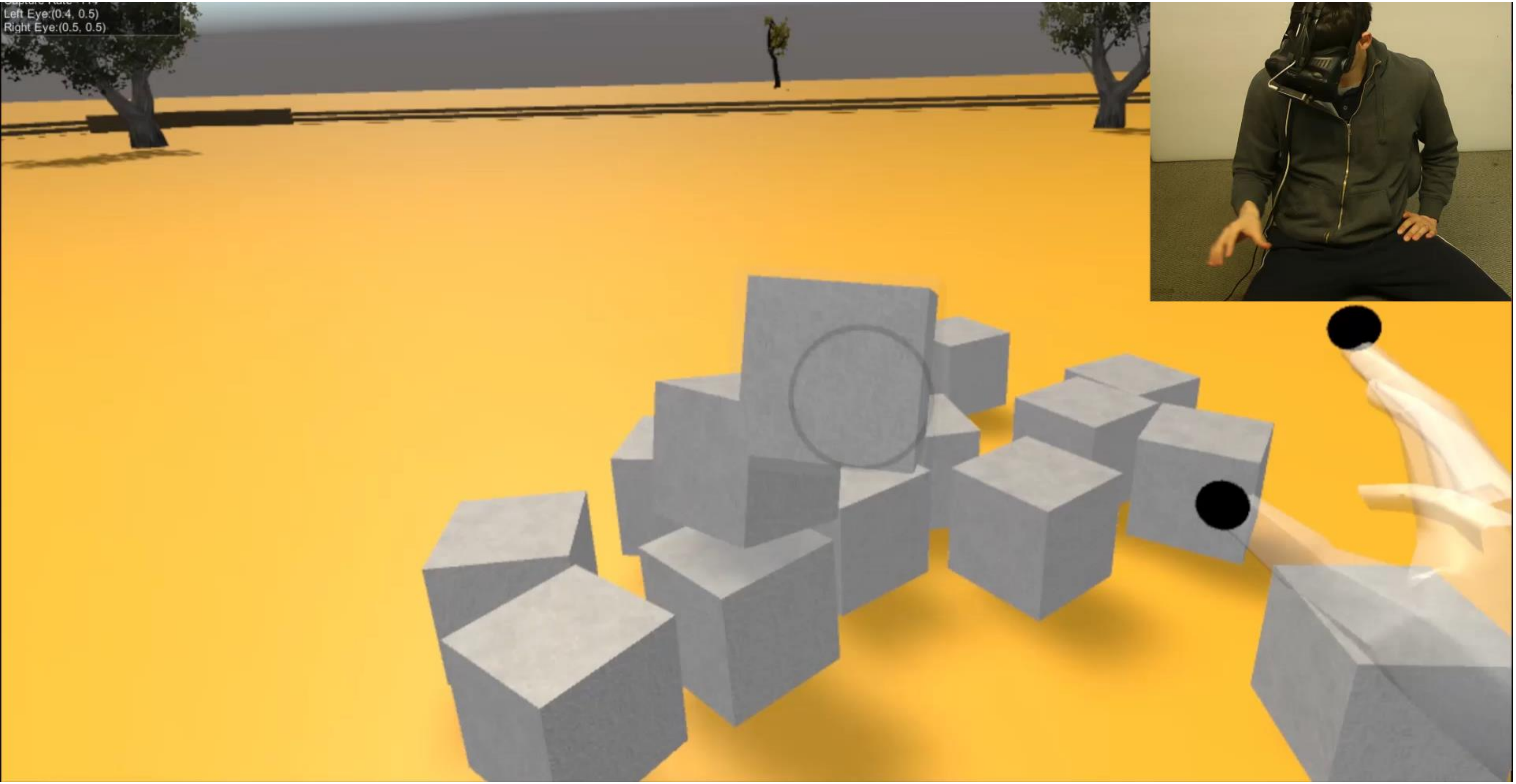
HTC VIVE + Leap Motion
+ Pupil eye tracker

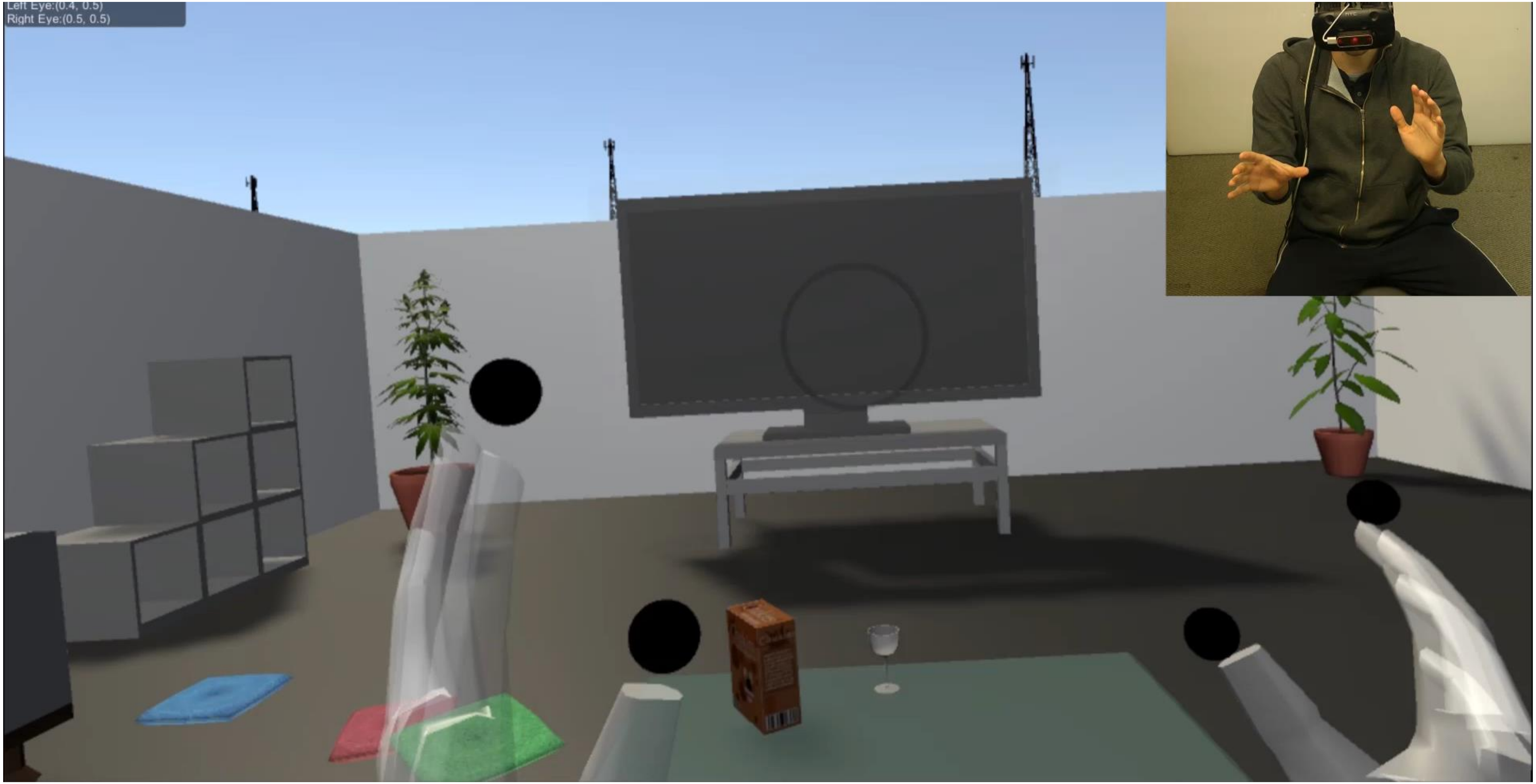


Virtual

Objects/scene in Unity 3D


Left Eye:(0.4, 0.5)
Right Eye:(0.5, 0.5)


Left Eye:(0.4, 0.5)
Right Eye:(0.5, 0.5)


Left Eye:(0.4, 0.5)
Right Eye:(0.5, 0.5)

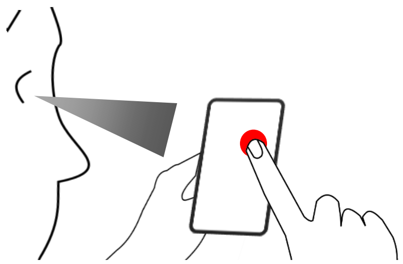
Using Visual attention in User Interfaces

Adapt UI to user.
Personalise, learn, enhance.

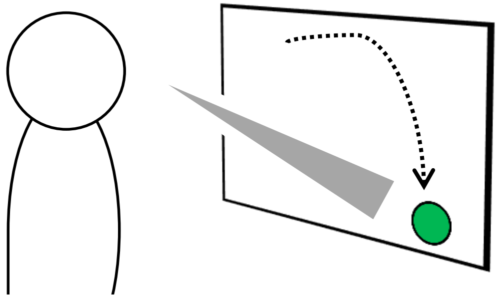
User controls UI with their eyes.
Select, use, manipulate.

Implicit

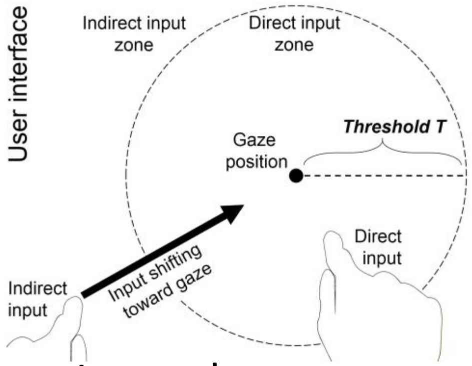
Explicit



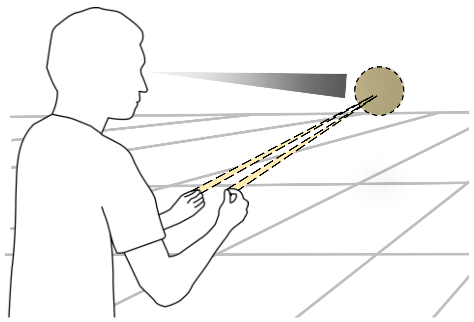
User performance modelling



Movement correlation & calibration



Input shortcuts



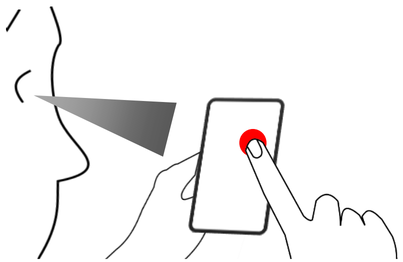
Gaze + Manual Input

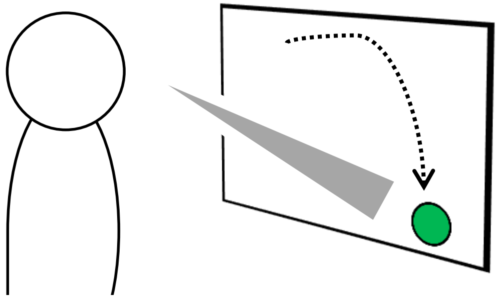
Using Visual Attention in User Interfaces

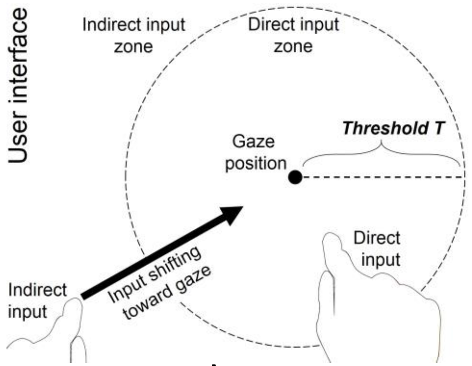
Adapt UI to user.
Personalise, learn, enhance.

User controls UI with their eyes.
Select, use, manipulate.

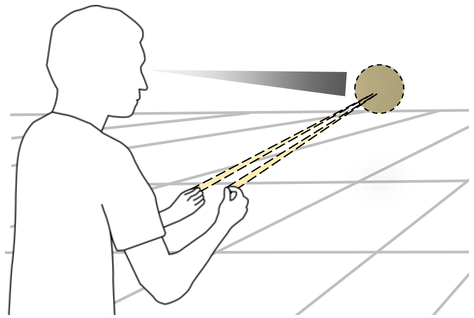
Implicit ⟷ Explicit


User performance modelling


Movement correlation & calibration



Input shortcuts


Gaze + Manual Input

Thank you! Any questions?

More information on kenpfeuffer.com